

به نام داناترین



دوره‌ی خلاقیت الگوریتمی و برنامه‌نویسی پایتون

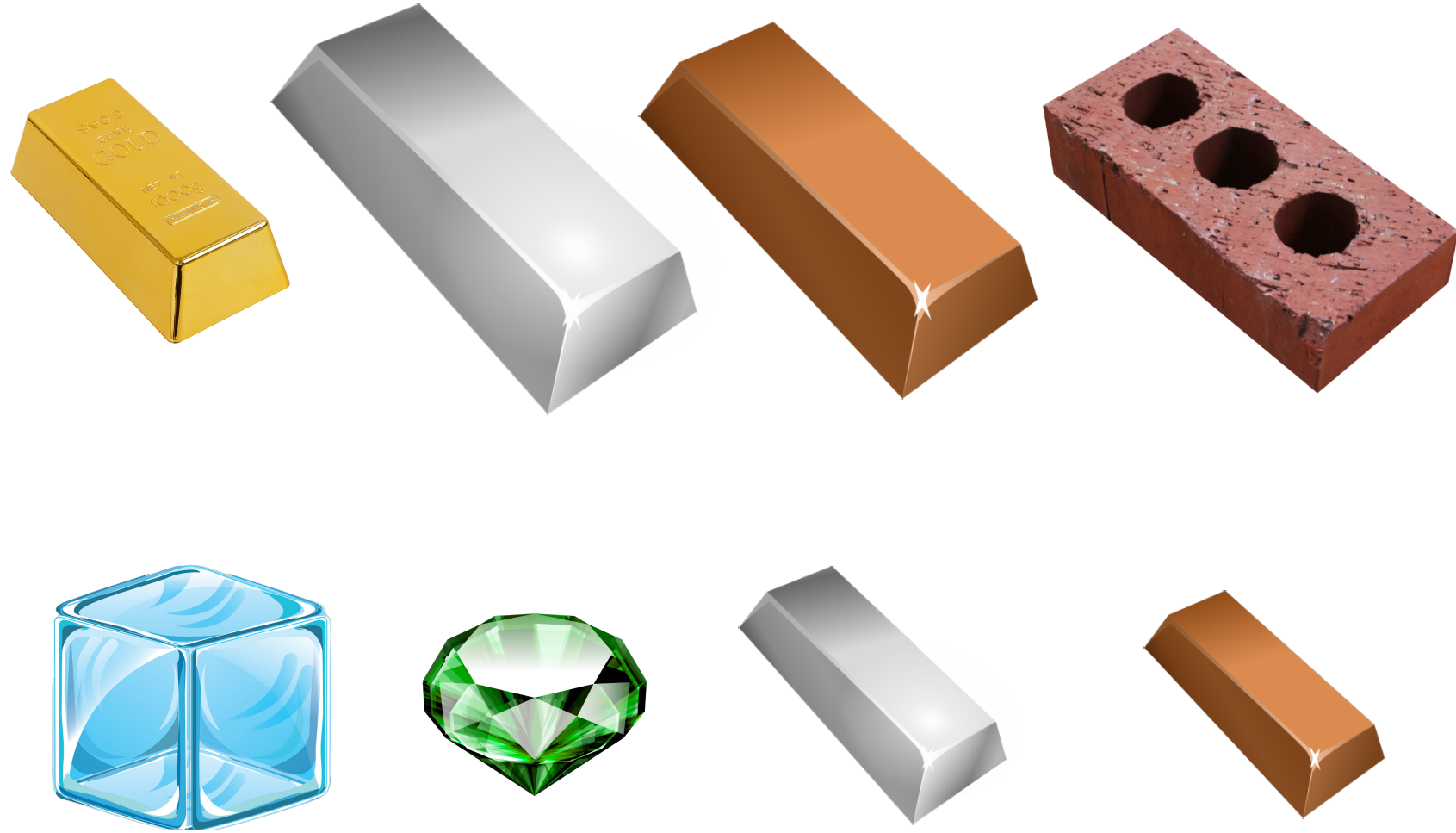
ماموریت کوله‌پشتی

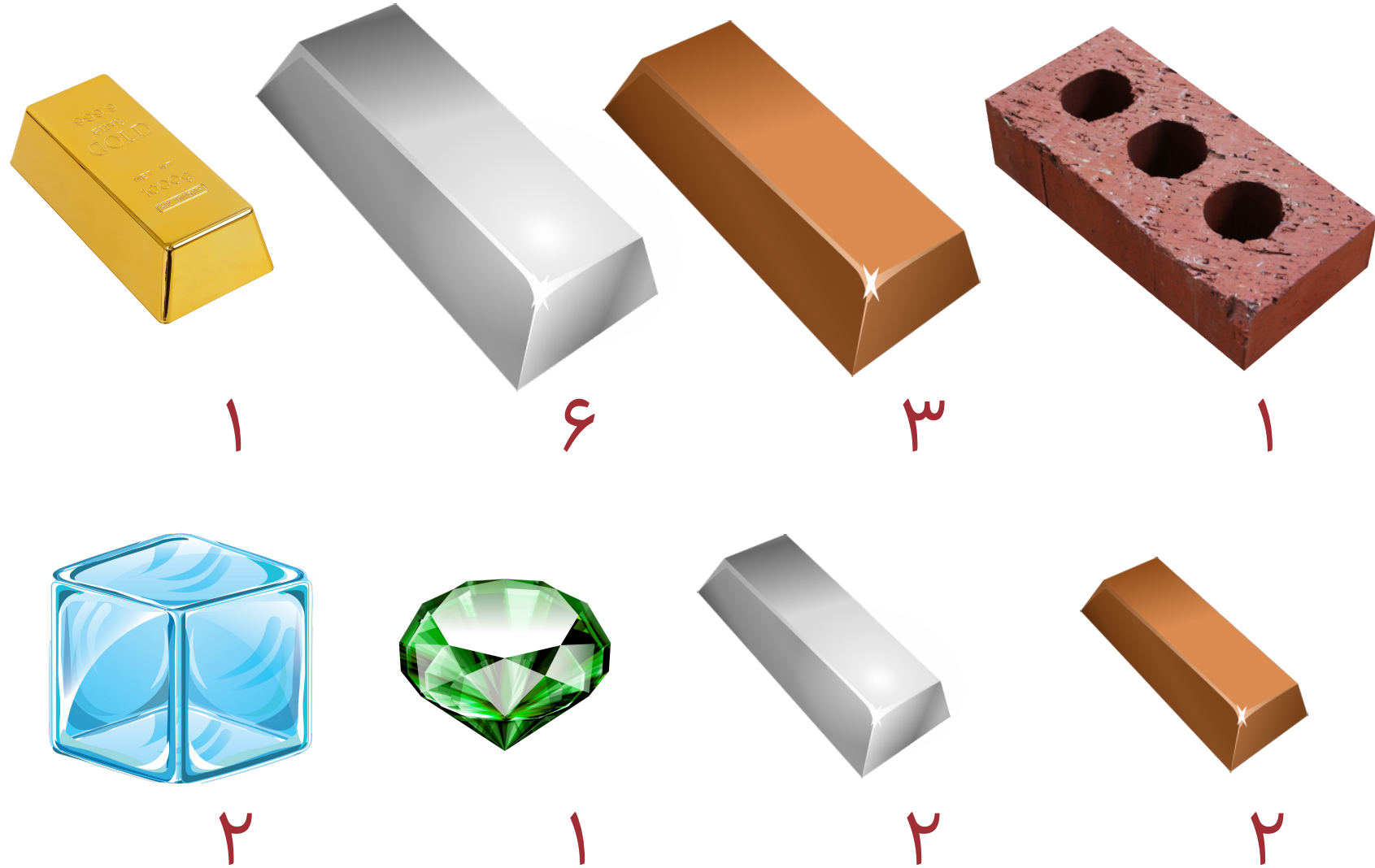
دانشکده‌ی مهندسی کامپیوتر دانشگاه صنعتی شریف

تابستان ۱۴۰۲



شمش، آجر، یخ، کریستال

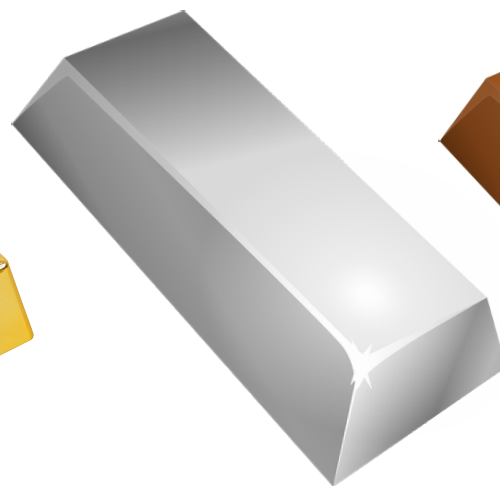




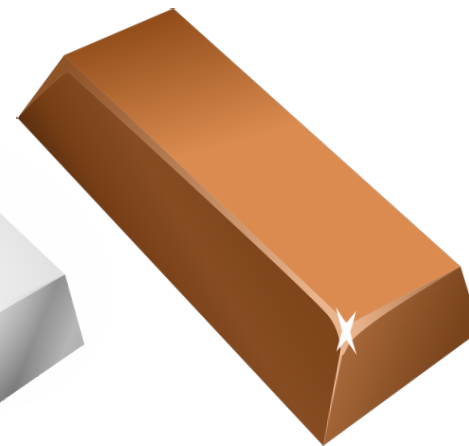
وزن، قیمت



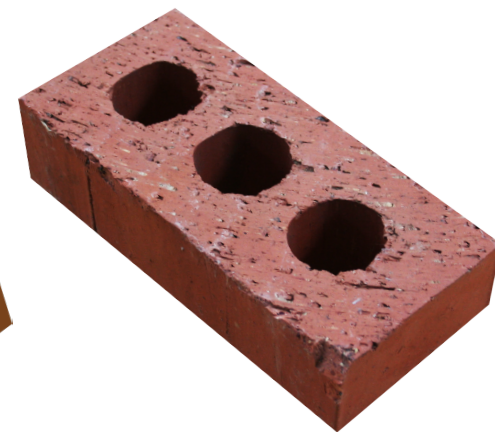
۱۰۰ ۱



۲۰ ۶



۱۰ ۳



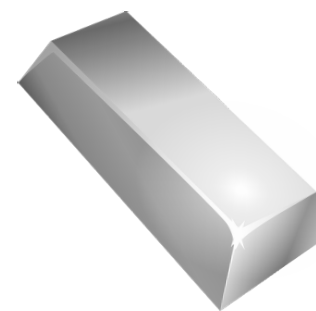
۱ ۱



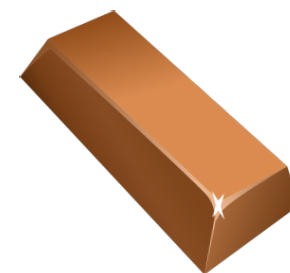
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

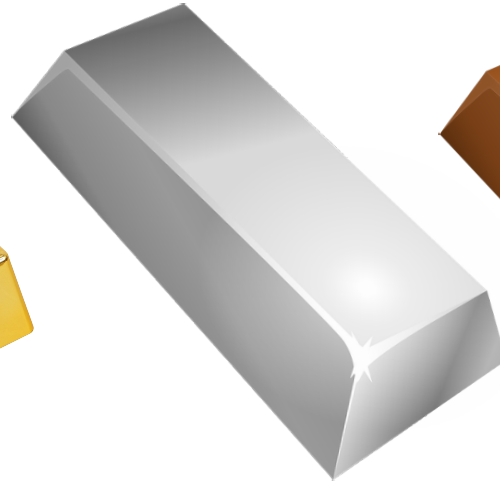
وزن، قیمت، وزن قابل حمل با کوله‌پشتی



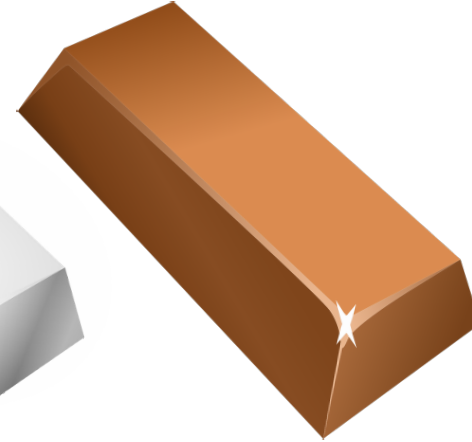
۹



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



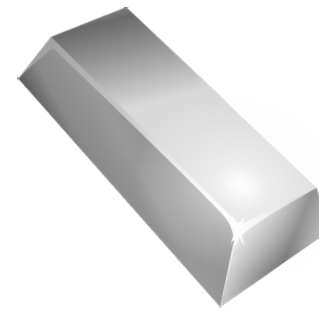
۱ ۱



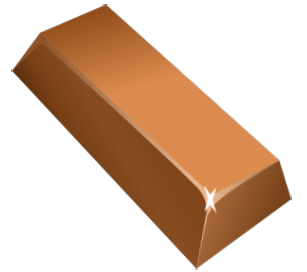
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

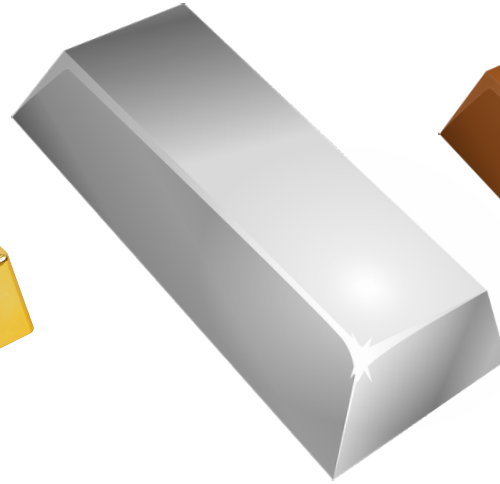
کدام کالاها را برداریم که بیشترین مجموع قیمت را داشته باشند؟



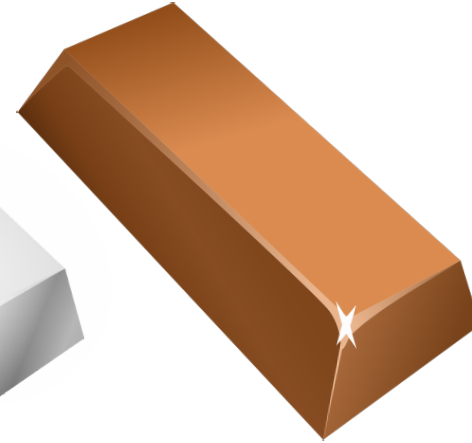
۹



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



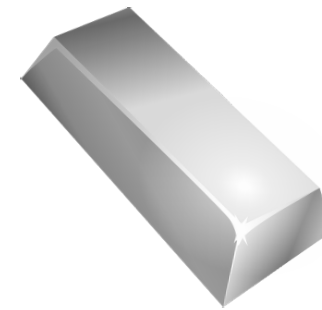
۱ ۱



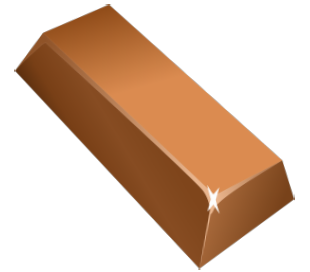
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

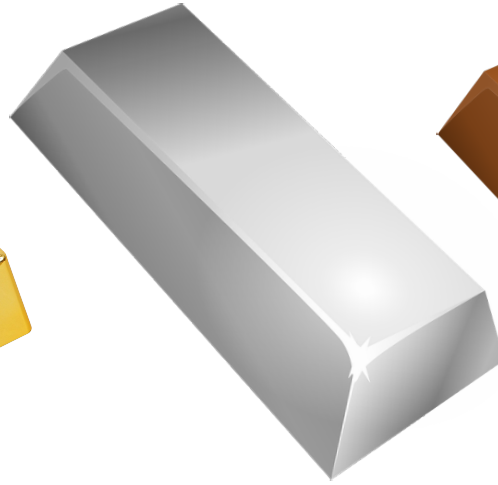
بیایید حریصانه زندگی کنیم (:



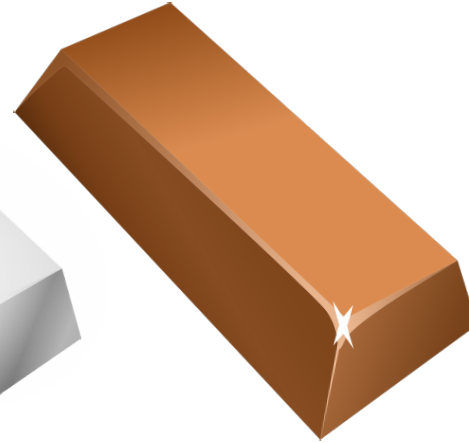
۹



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



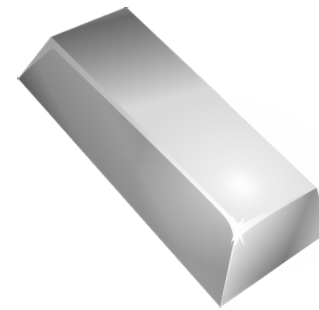
۱ ۱



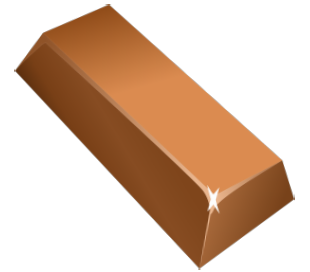
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

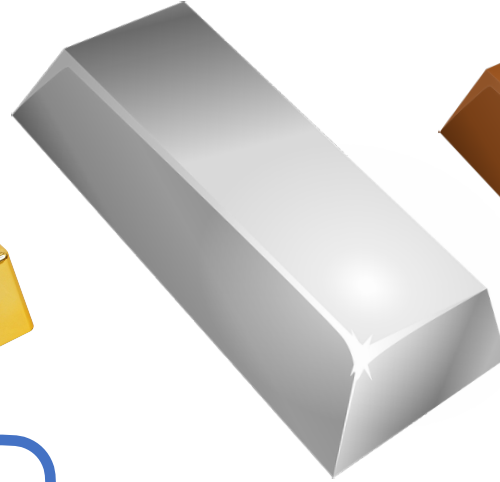
بیاید حریصانه زندگی کنیم (:



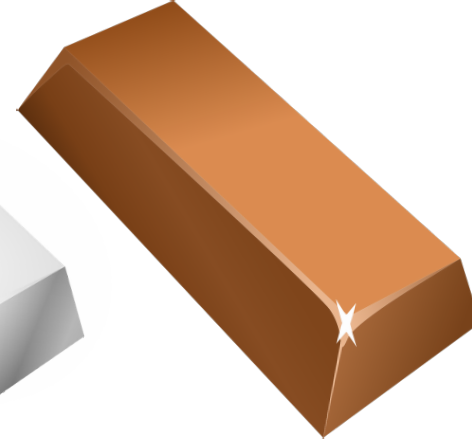
۹



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



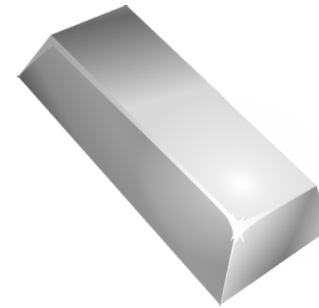
۱ ۱



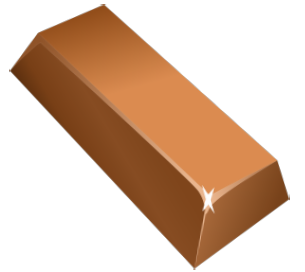
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

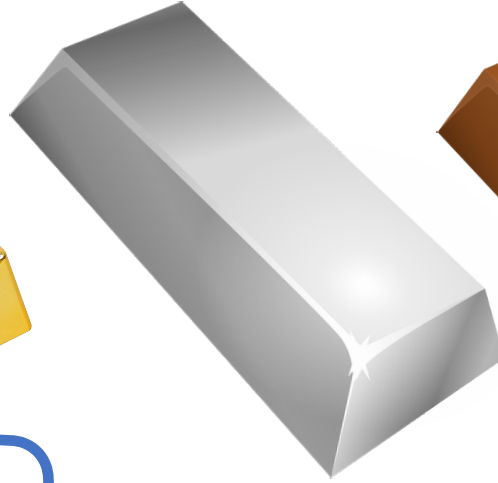
بیا یاد حریصانه زندگی کنیم (:



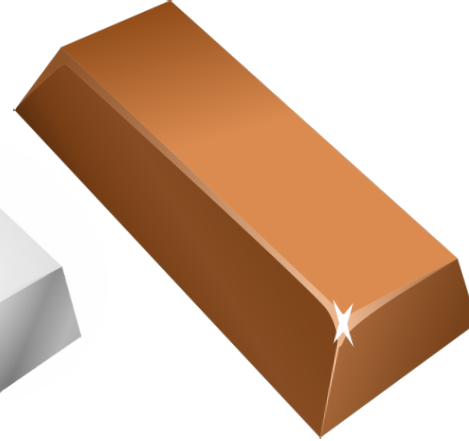
۱۰۰ ۸



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



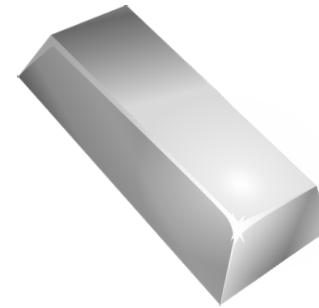
۱ ۱



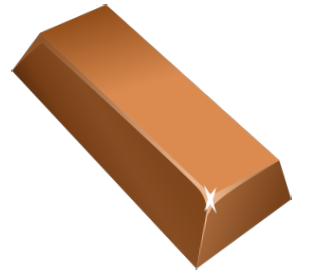
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

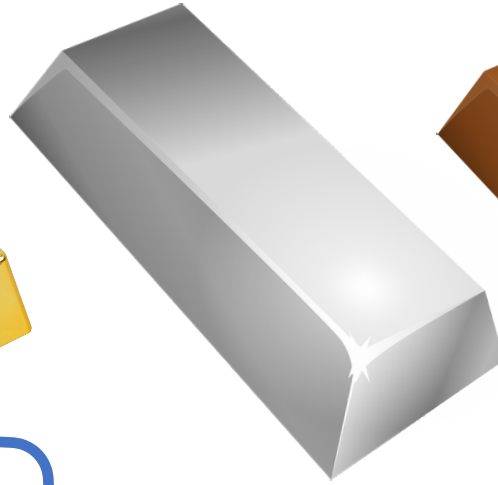
بیا یاد حریصانه زندگی کنیم (:



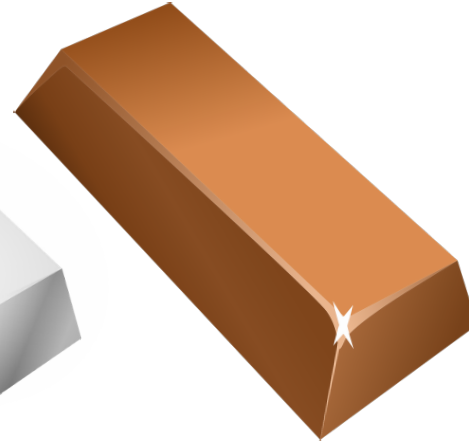
۱۰۰ ۸



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



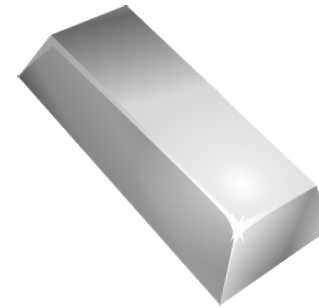
۱ ۱



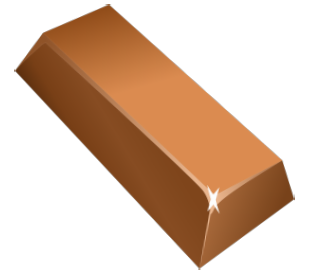
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

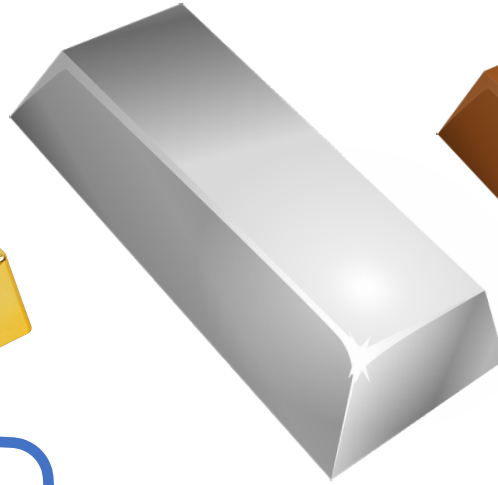
بیا یاد حریصانه زندگی کنیم (:



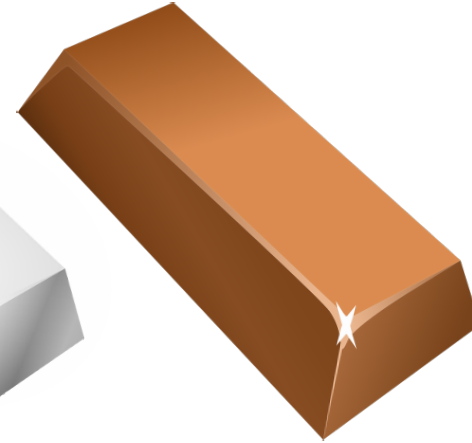
۱۸۰ ۷



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



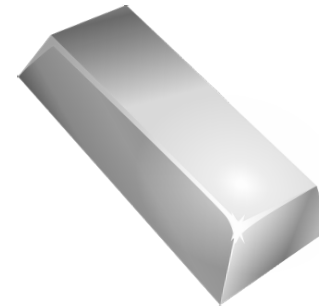
۱ ۱



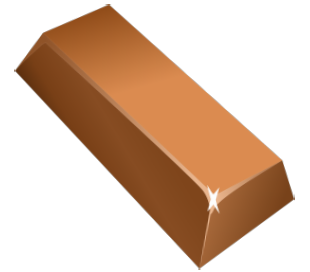
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

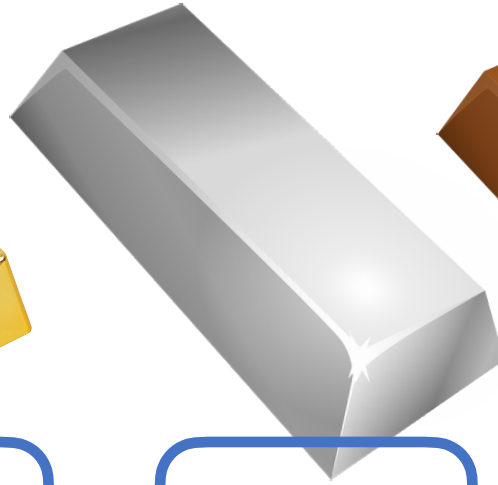
بیایید حریصانه زندگی کنیم (:



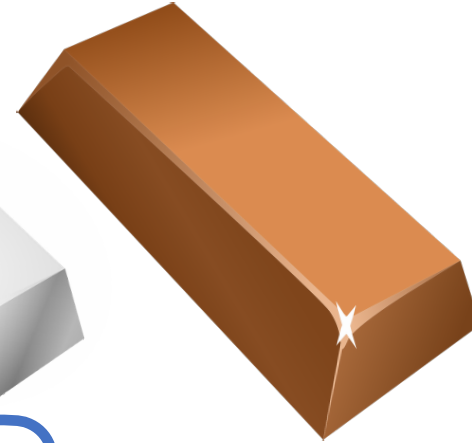
۱۸۰ ۷



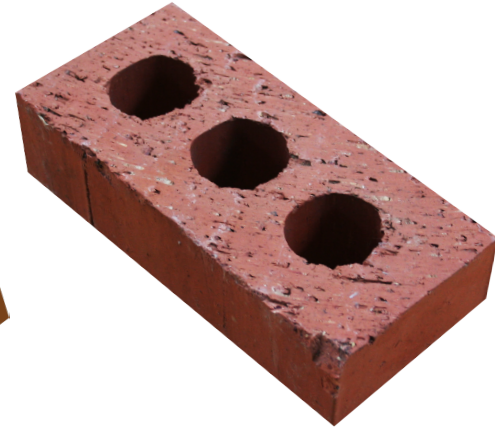
۱۰۰ ۱



۲۰ ۶



۱۰ ۳



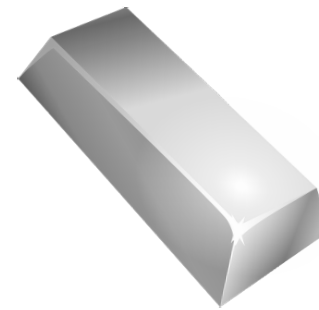
۱ ۱



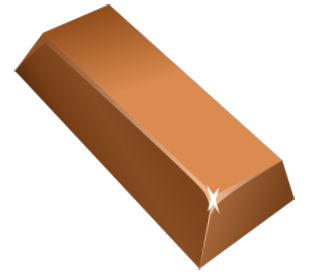
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

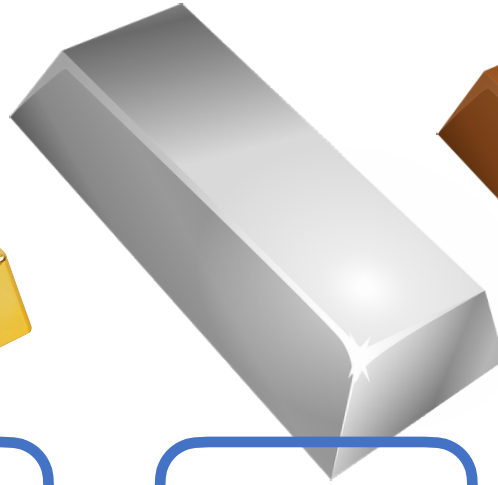
بیا یاد حریصانه زندگی کنیم (:



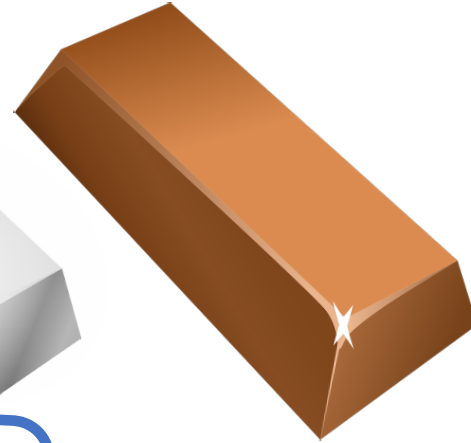
۲۰۰ ۱



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



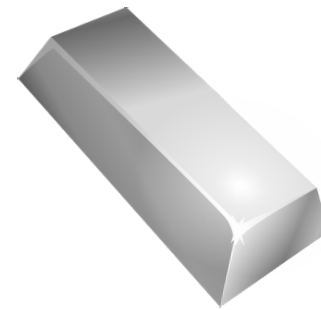
۱ ۱



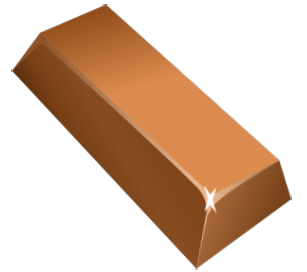
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

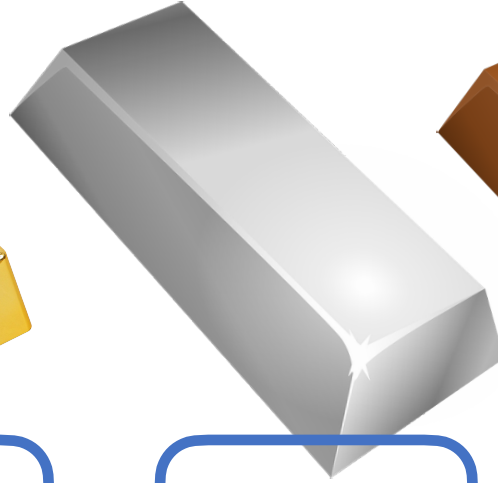
بیا یاد حریصانه زندگی کنیم (:



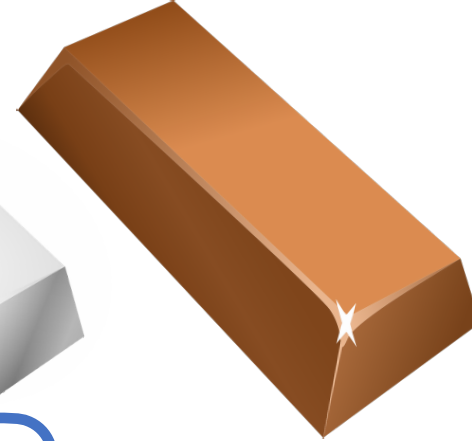
۲۰۰ ۱



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



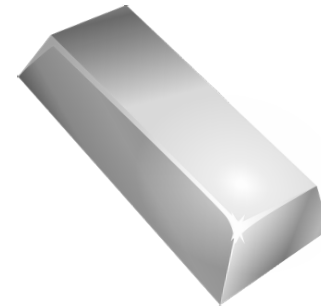
۱ ۱



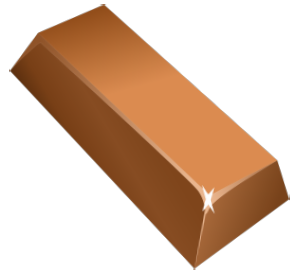
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

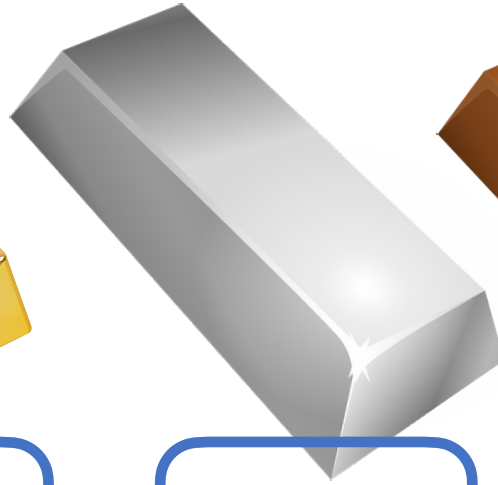
بیا یاد حریصانه زندگی کنیم (:



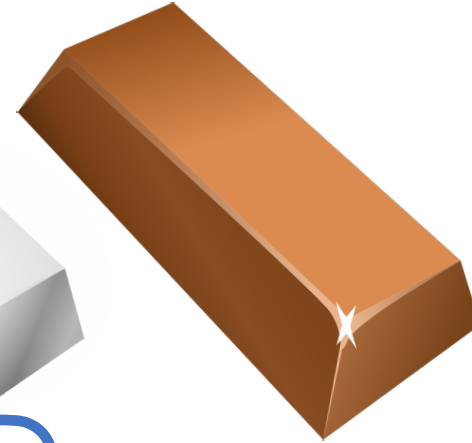
۲۰۱ ۰



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



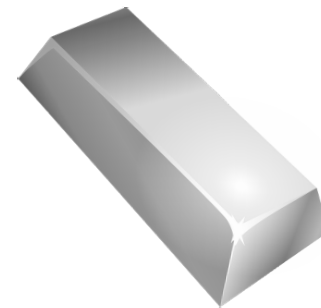
۱ ۱



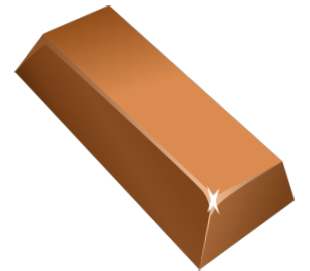
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

آیا انتخاب **گران‌ترین** کالای قابل حمل، همیشه پاسخ بهینه را پیدا می‌کند؟

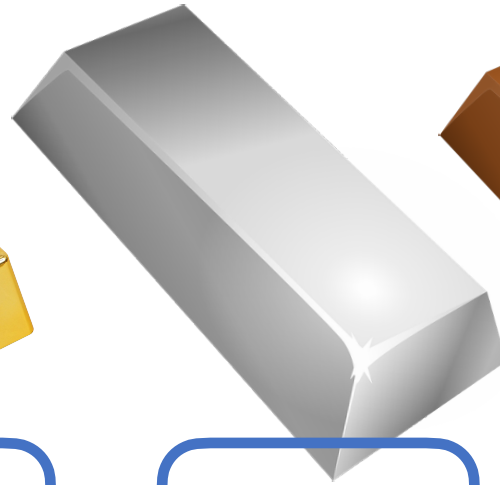
آیا انتخاب گران‌ترین کالای قابل حمل، همیشه پاسخ بهینه را پیدا می‌کند؟



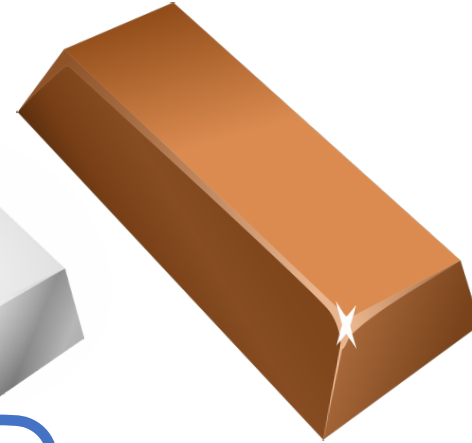
۲۰۱ ۰



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



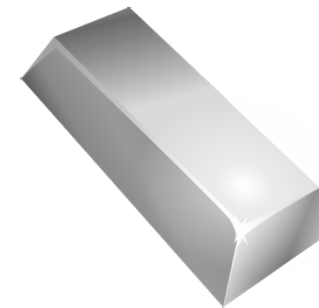
۱ ۱



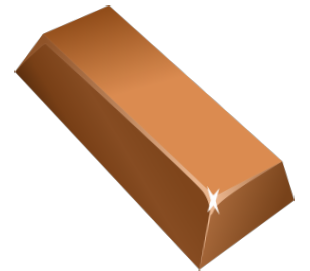
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

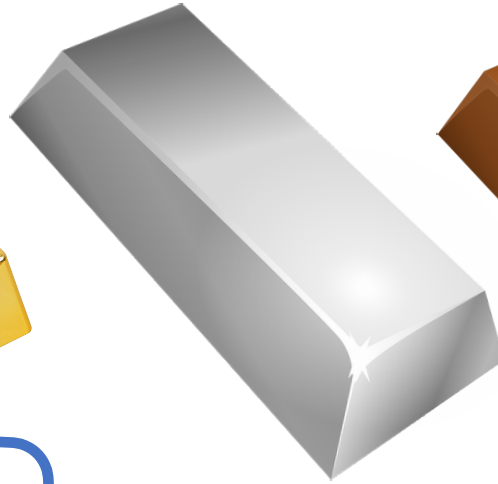
پاسخ بهتر...



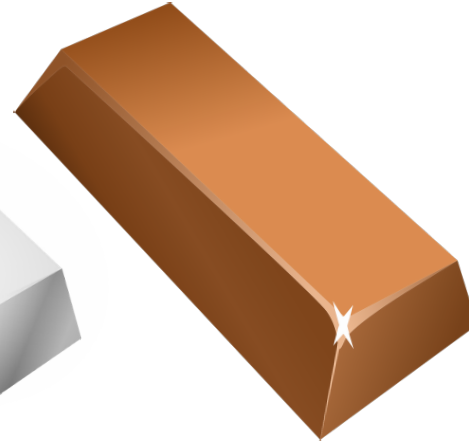
۱۸۰ ۷



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



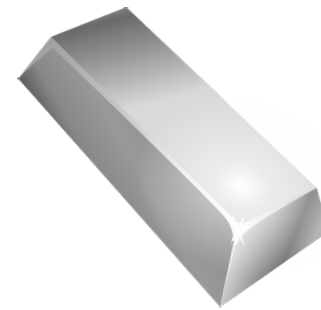
۱ ۱



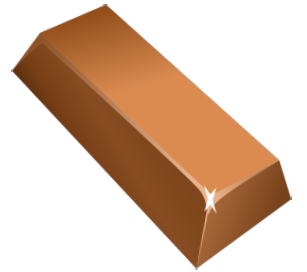
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

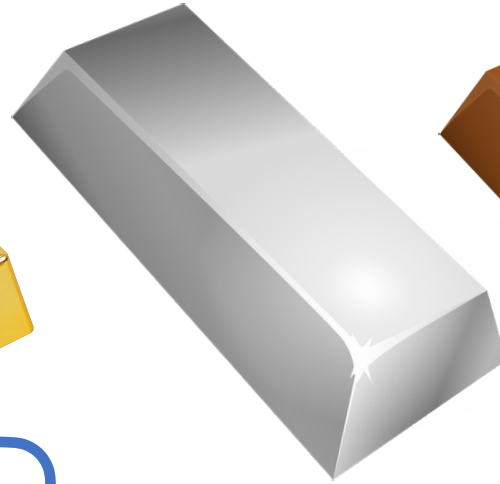
پاسخ بهتر...



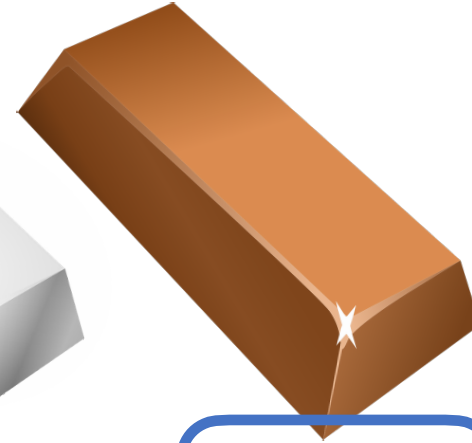
۱۸۰ ۷



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



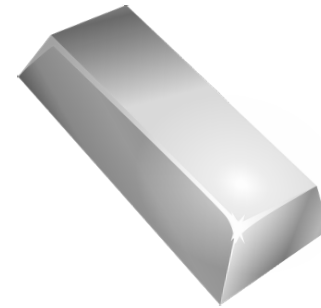
۱ ۱



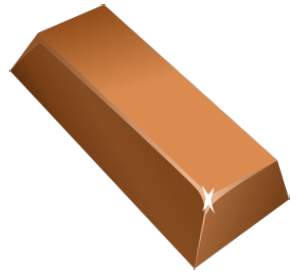
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

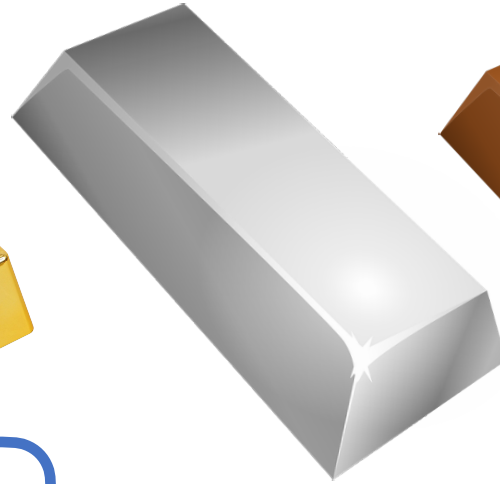
پاسخ بهتر...



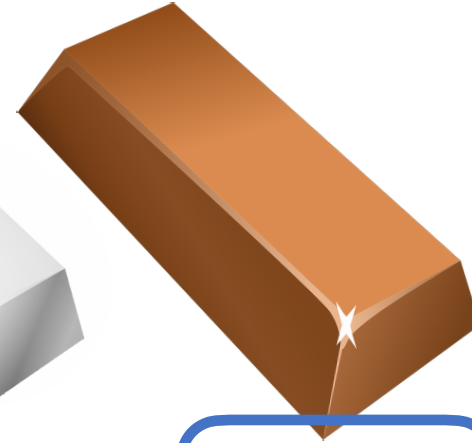
۱۹۰ ۴



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



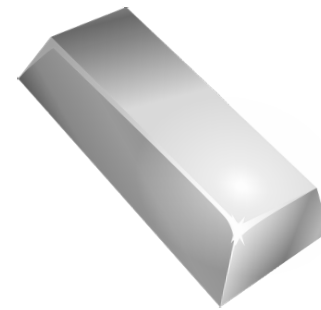
۱ ۱



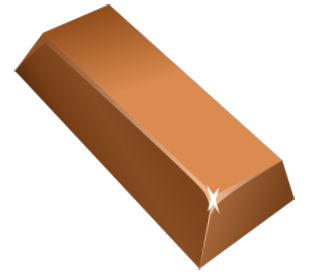
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

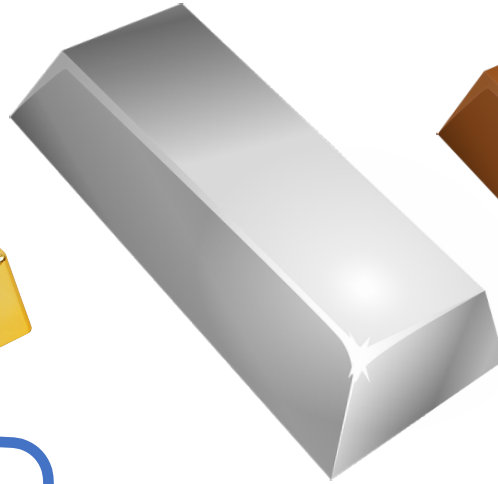
پاسخ بهتر...



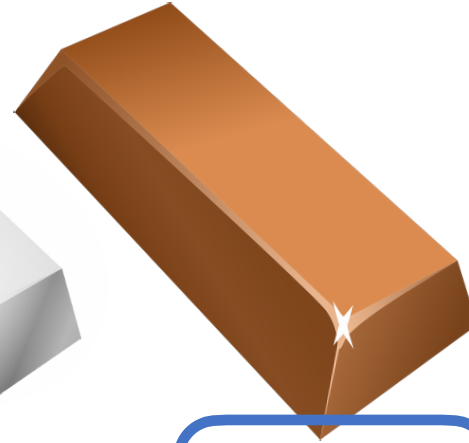
۱۹۰ ۴



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



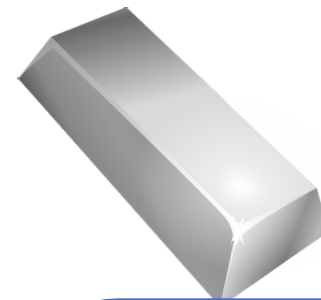
۱ ۱



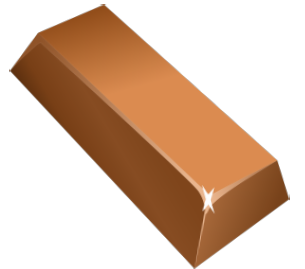
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

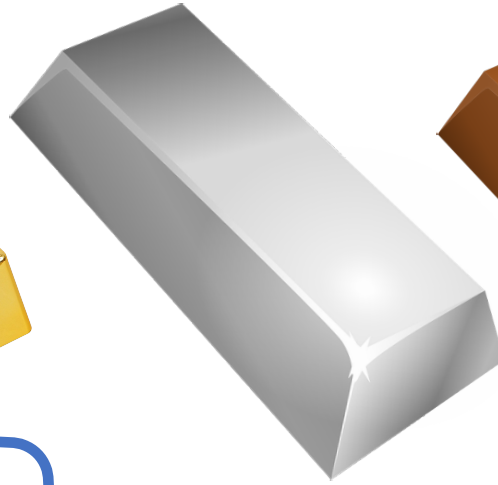
پاسخ بهتر...



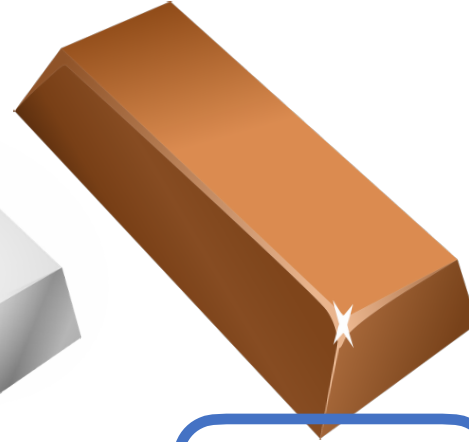
۲۰۰ ۲



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



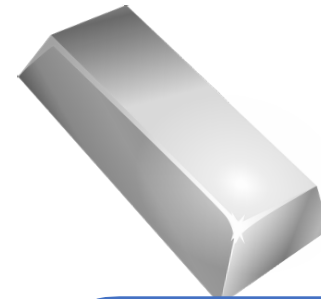
۱ ۱



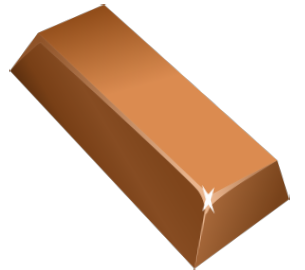
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

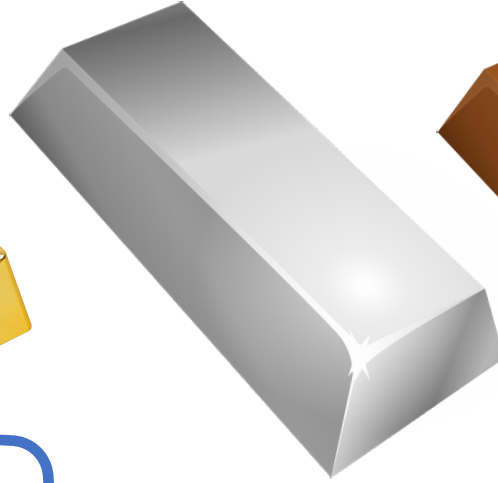
پاسخ بهتر...



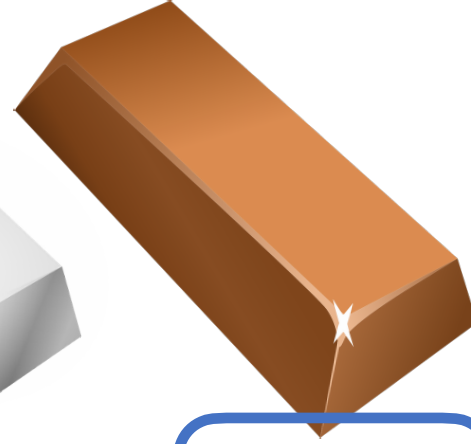
۲۰۰ ۲



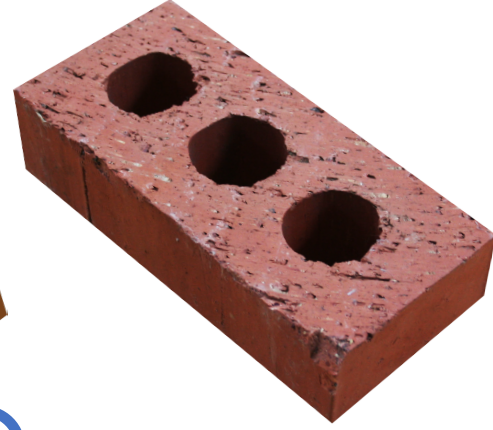
۱۰۰ ۱



۲۰ ۶



۱۰ ۳



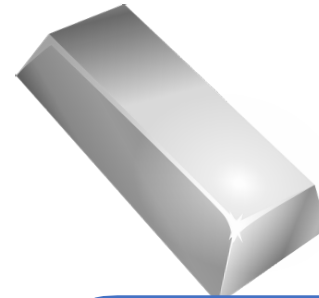
۱ ۱



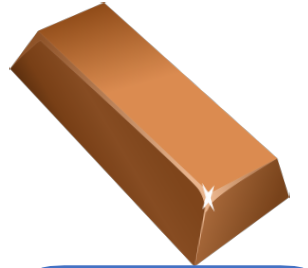
۱ ۲



۸۰ ۱



۱۰ ۲



۵ ۲

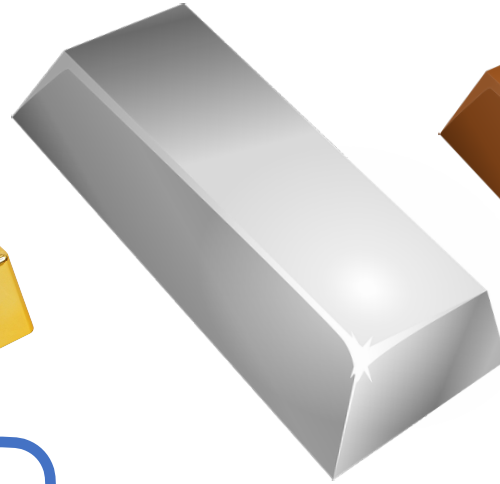
پاسخ بهتر...



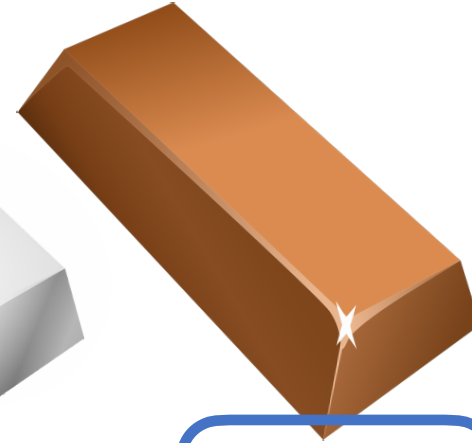
۲۰۵۰



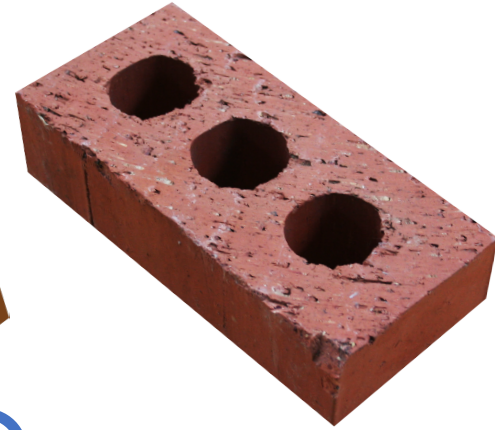
۱۰۰ ۱



۲۰ ۶



۱۰ ۳



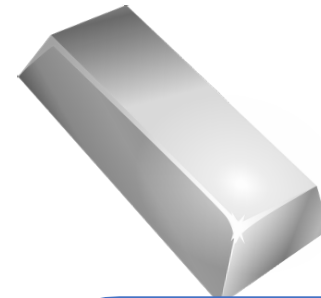
۱ ۱



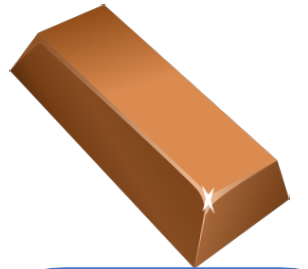
۱ ۲



۸۰ ۱



۱۰ ۲



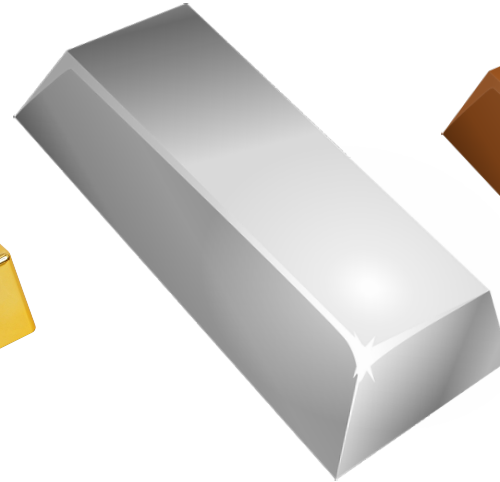
۵ ۲

آیا انتخاب کالایی که بیشترین نسبت قیمت به وزن دارد، همیشه پاسخ بهینه را پیدا می‌کند؟

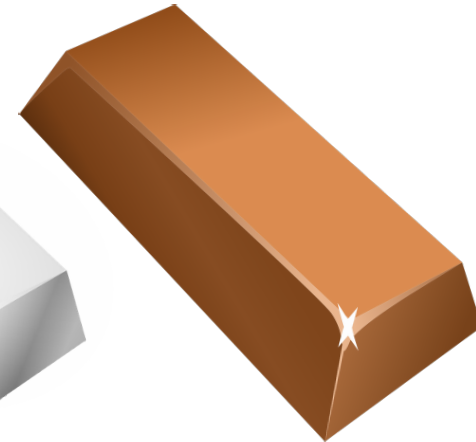
آیا انتخاب کالایی که بیشترین نسبت قیمت به وزن دارد، همیشه پاسخ بهینه را پیدا می‌کند؟



۱۰۰ ۱۰



۵۵ ۶



۵۵ ۶

○ ۱۲

آیا انتخاب کالایی که بیشترین نسبت قیمت به وزن دارد، همیشه پاسخ بهینه را پیدا می‌کند؟

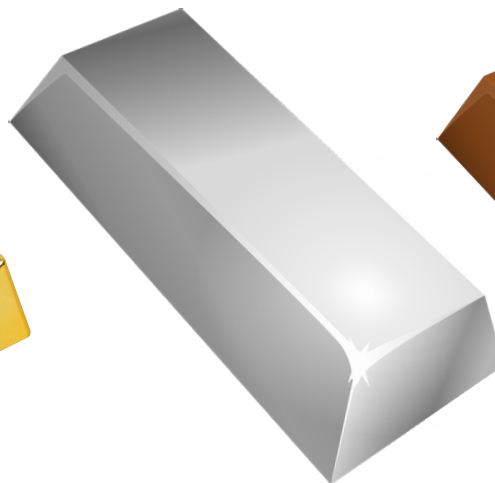


○ ۱۲



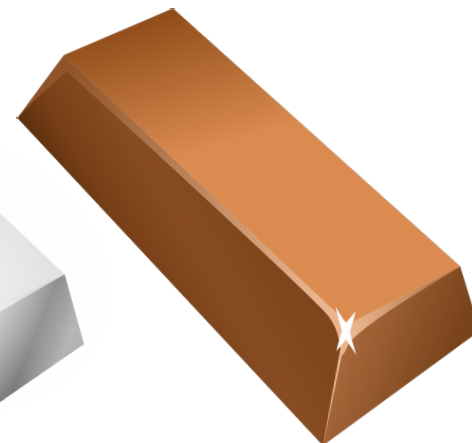
۱۰۰ ۱۰

۱۰



۵۵ ۶

۹.۲



۵۵ ۶

۹.۲

قیمت به وزن

آیا انتخاب کالایی که بیشترین نسبت قیمت به وزن دارد، همیشه پاسخ بهینه را پیدا می‌کند؟

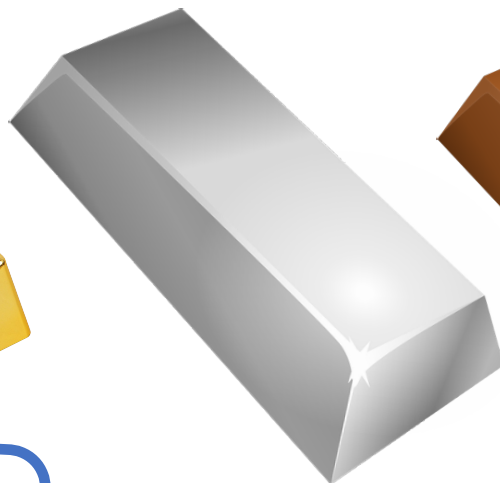


○ ۱۲



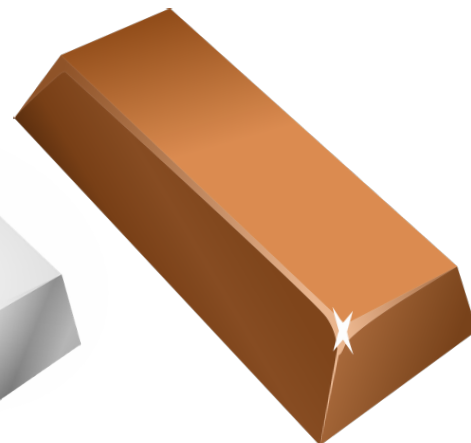
۱۰۰ ۱۰

۱۰



۵۵ ۶

۹.۲



۵۵ ۶

۹.۲

قیمت به وزن

آیا انتخاب کالایی که بیشترین نسبت قیمت به وزن دارد، همیشه پاسخ بهینه را پیدا می‌کند؟

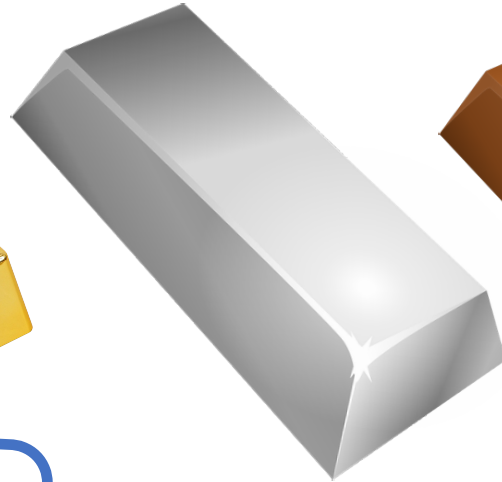


۱۰۰ ۲



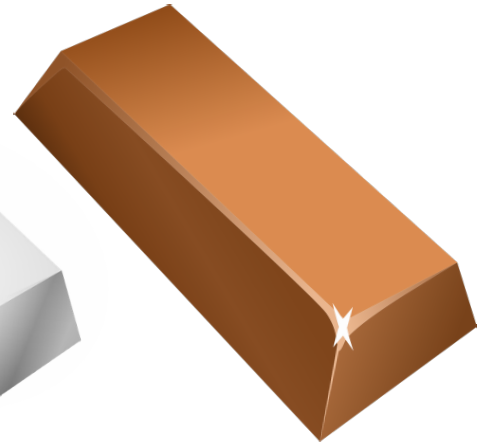
۱۰۰ ۱۰

۱۰



۵۵ ۶

۹.۲



۵۵ ۶

۹.۲

قیمت به وزن

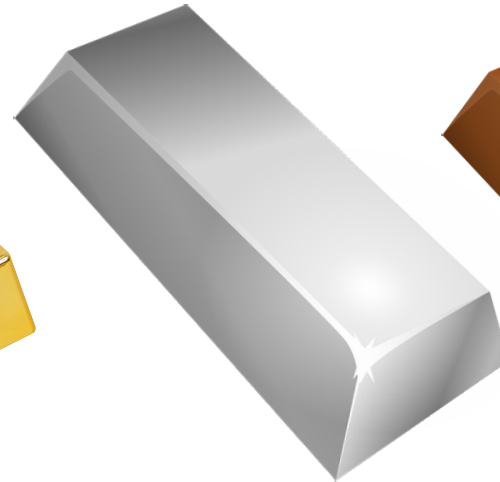


○ ۱۲



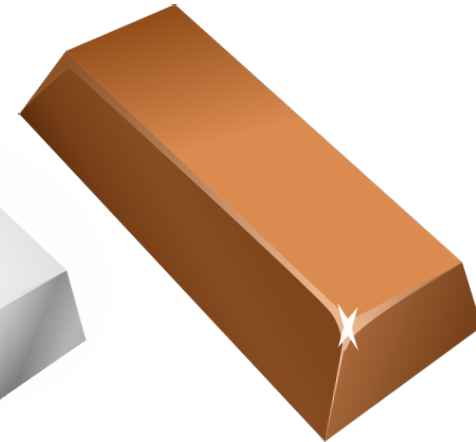
۱۰۰ ۱۰

۱۰



۵۵ ۶

۹.۲



۵۵ ۶

۹.۲

قیمت به وزن

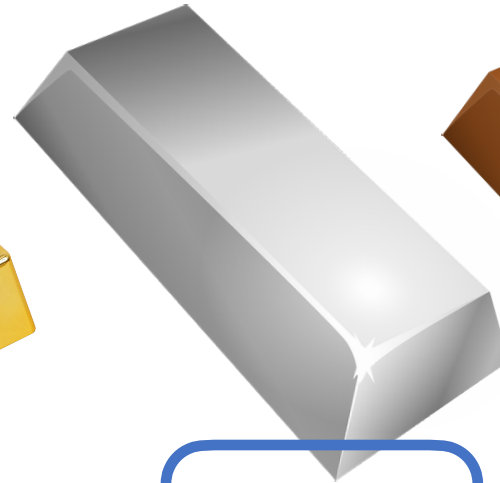


○ ۱۲



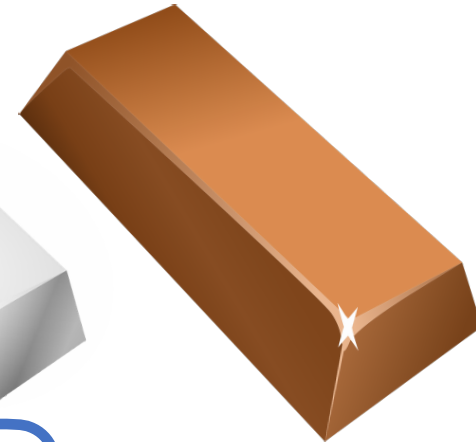
۱۰۰ ۱۰

۱۰



۵۵ ۶

۹.۲



۵۵ ۶

۹.۲

قیمت به وزن

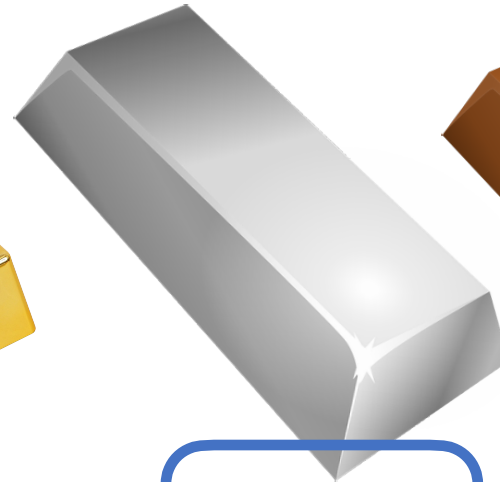


۵۵ ۶



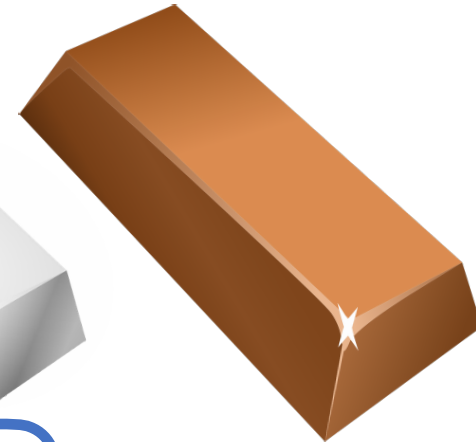
۱۰۰ ۱۰

۱۰



۵۵ ۶

۹.۲



۵۵ ۶

۹.۲

قیمت به وزن

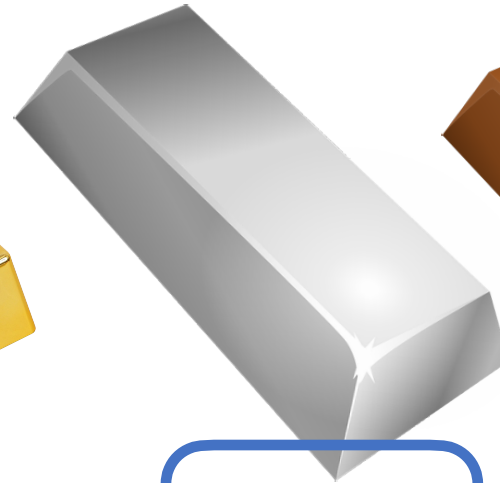


۵۵ ۶



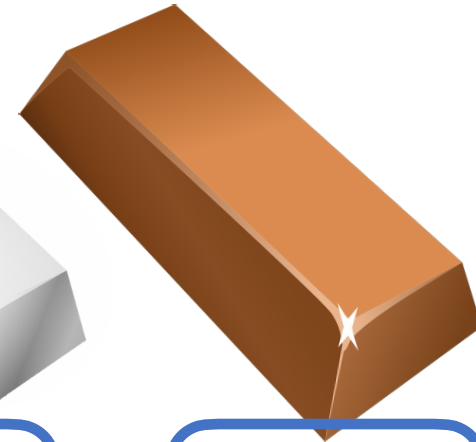
۱۰۰ ۱۰

۱۰



۵۵ ۶

۹.۲



۵۵ ۶

۹.۲

قیمت به وزن

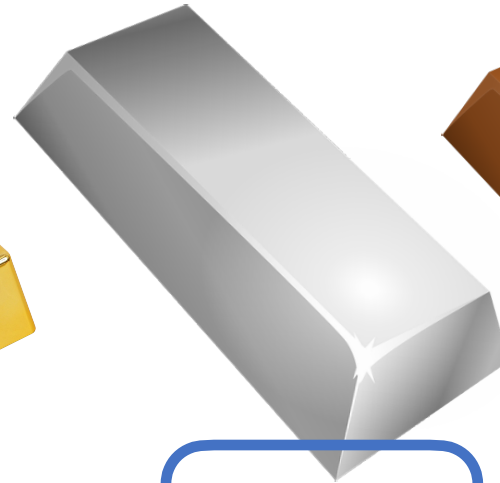


۱۱۰ ◦



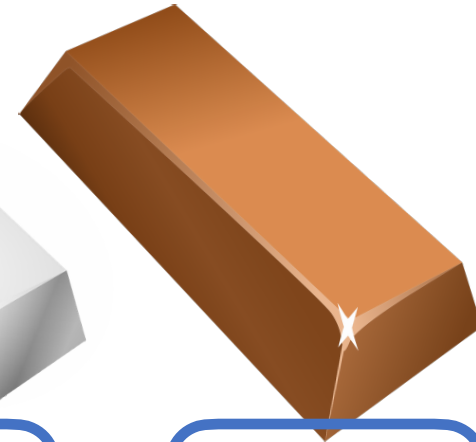
۱۰۰ ۱۰

۱۰



۵۵ ۶

۹.۲

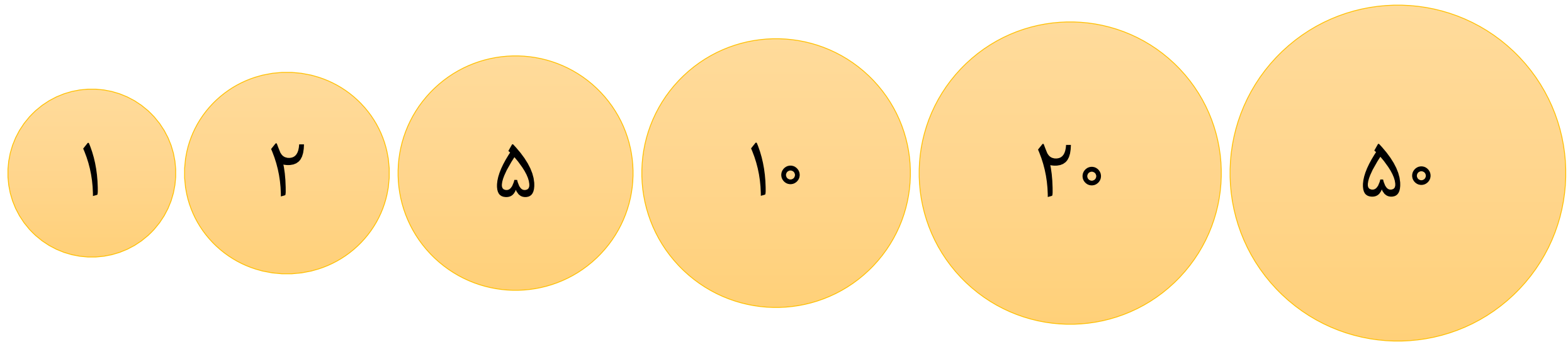


۵۵ ۶

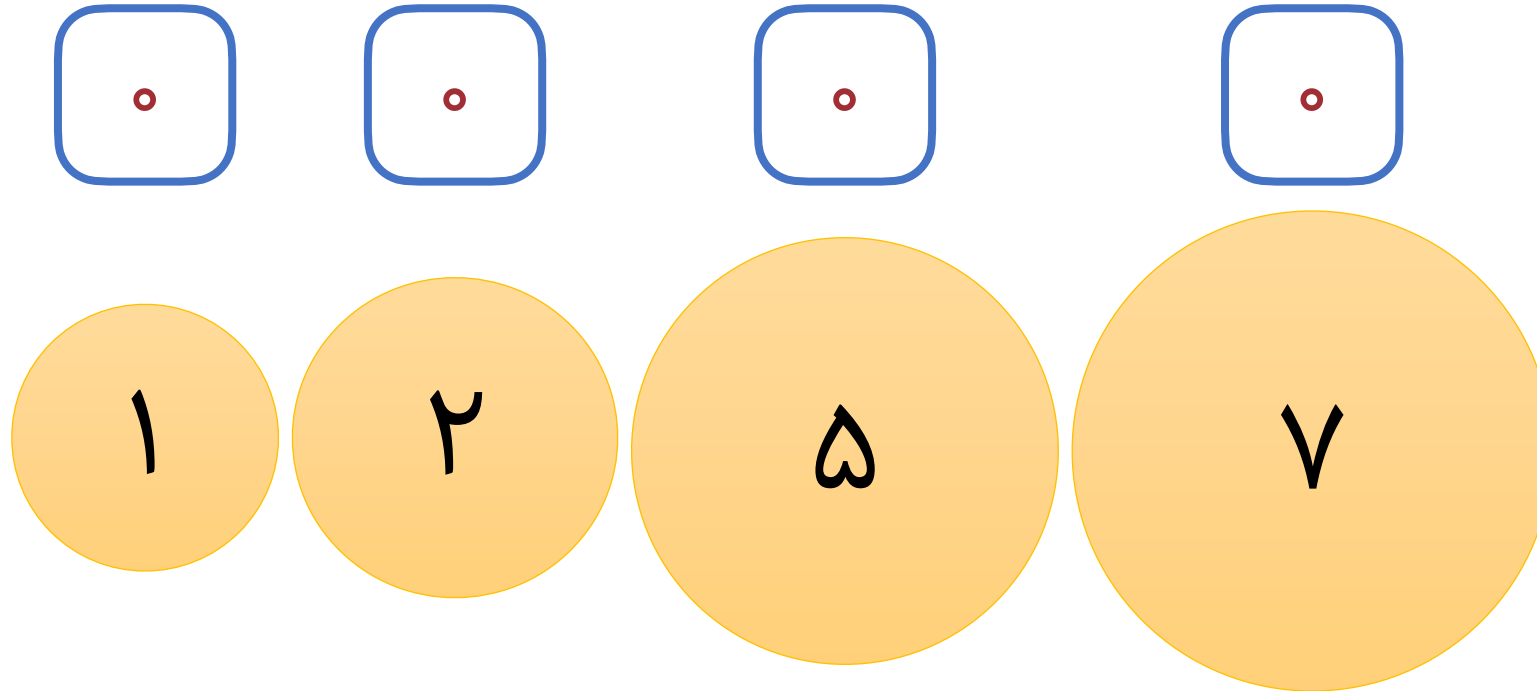
۹.۲

قیمت به وزن

الگوریتم حریمانه این مساله را نمی‌تواند حل کند!



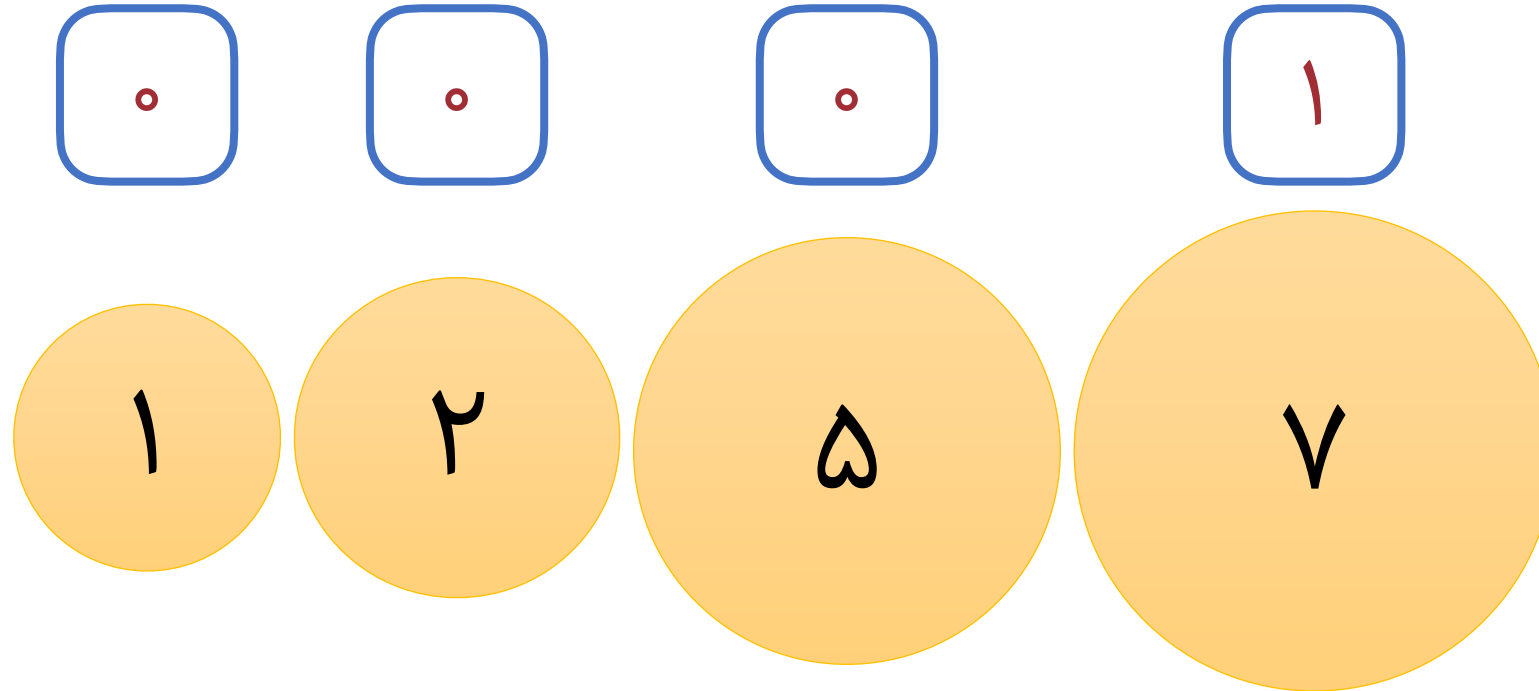
آیا الگوریتم حریمانه جواب بهینه را پیدا می‌کند؟



باقیمانده‌ی پول مشتری

۱۰

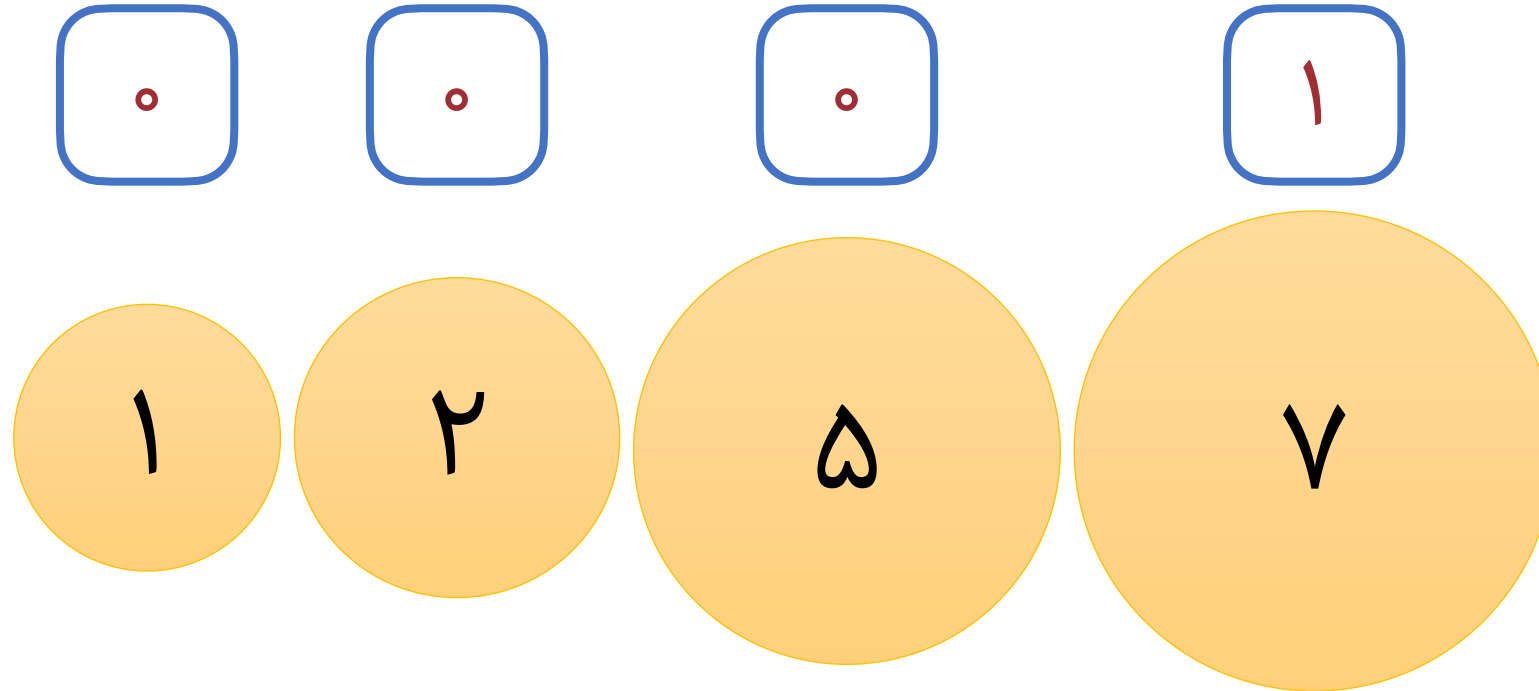
آیا الگوریتم حریمانه جواب بهینه را پیدا می‌کند؟



باقیمانده‌ی پول مشتری

۱۰

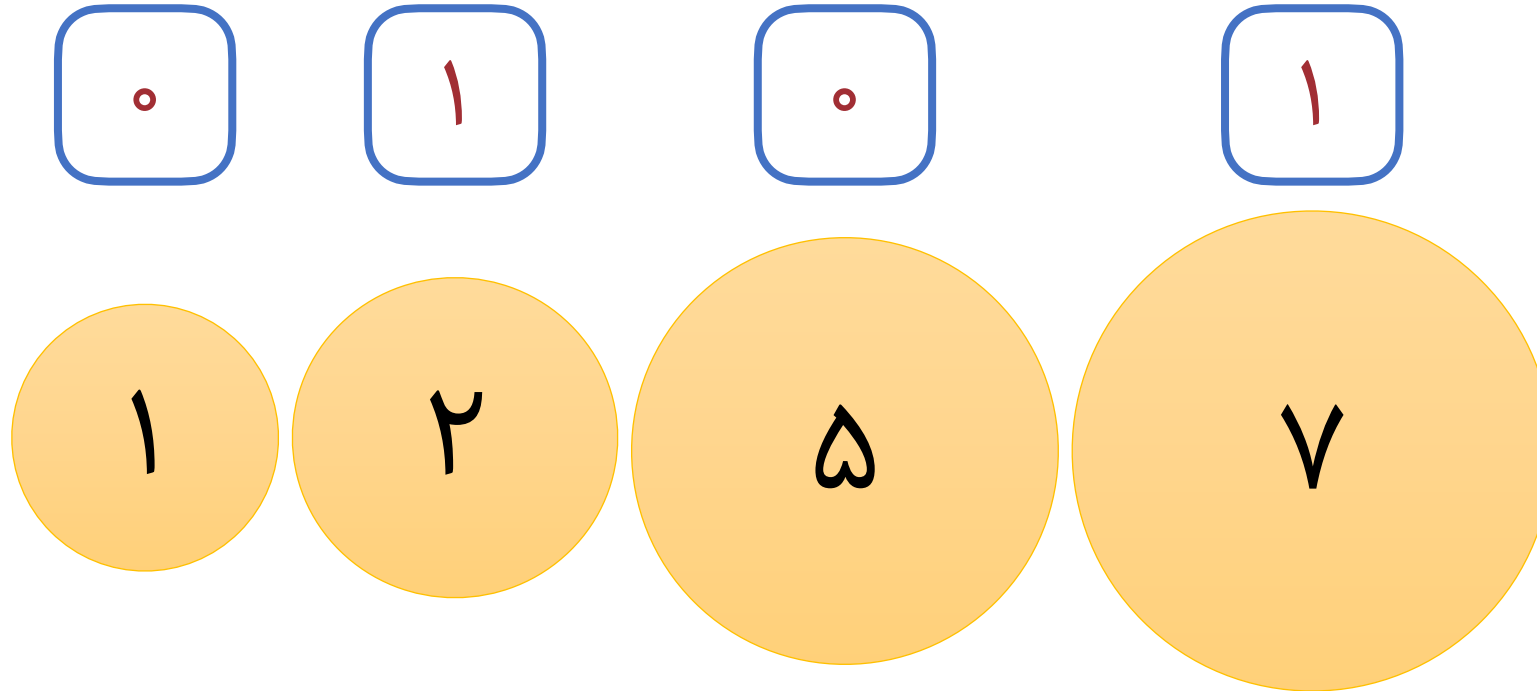
آیا الگوریتم حریمانه جواب بهینه را پیدا می‌کند؟



باقیمانده‌ی پول مشتری

۳

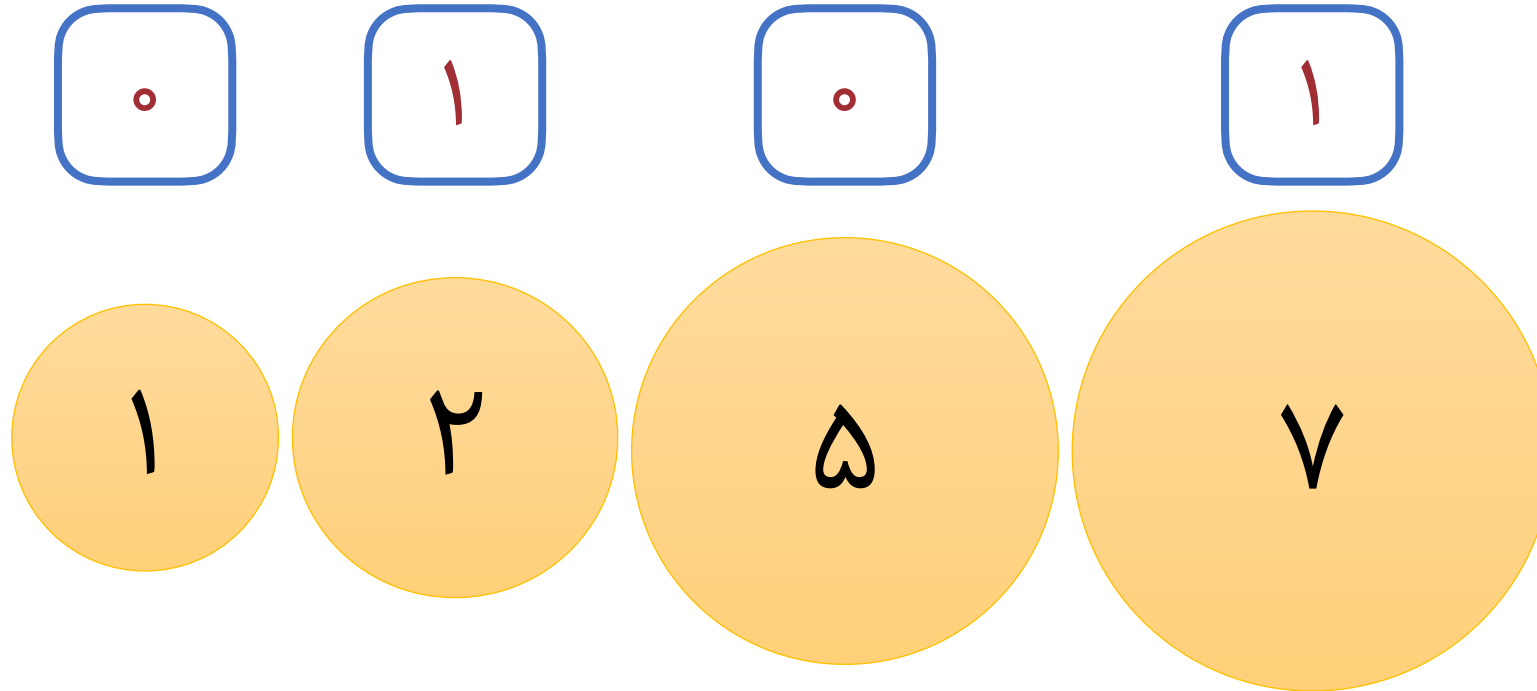
آیا الگوریتم حریمانه جواب بهینه را پیدا می‌کند؟



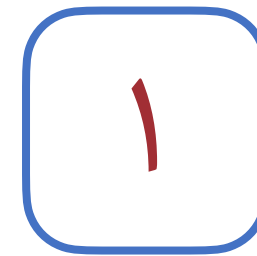
باقیمانده‌ی پول مشتری

۳

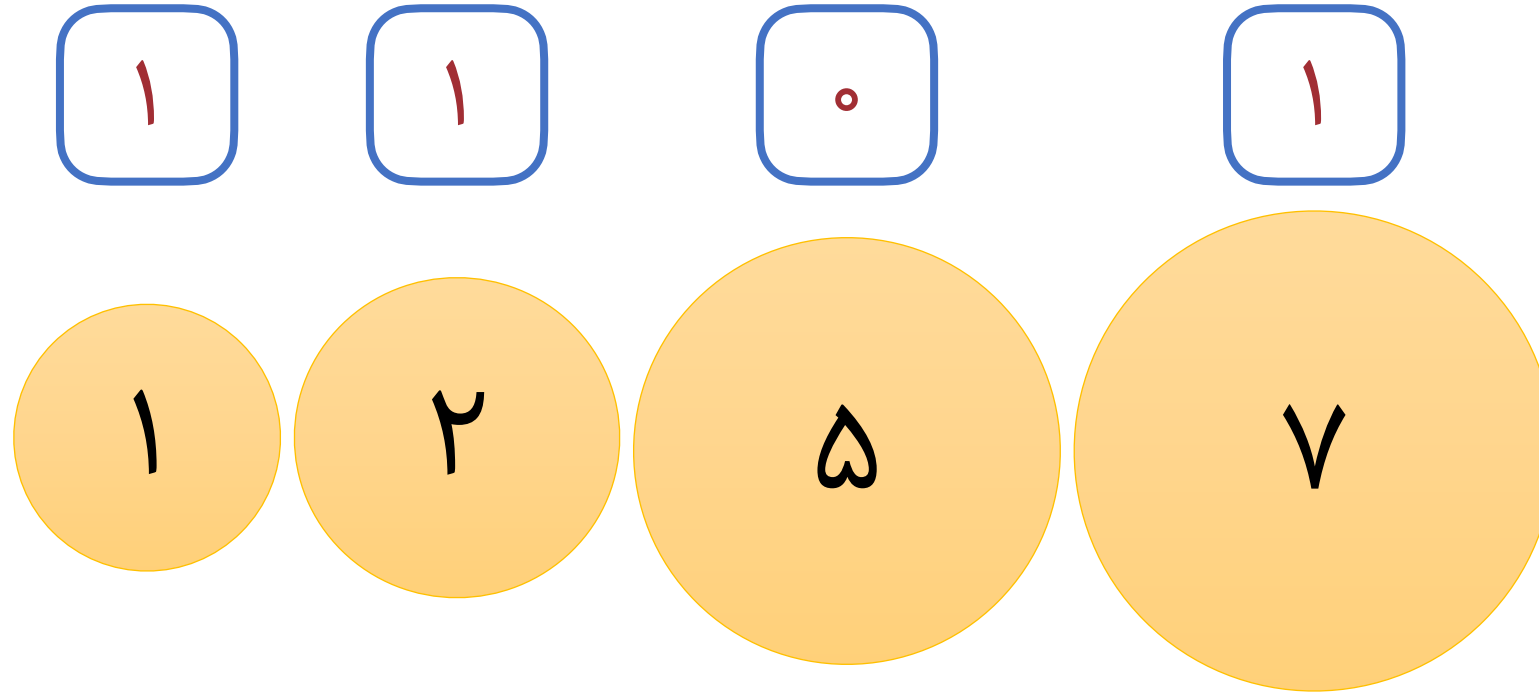
آیا الگوریتم حریصانه جواب بهینه را پیدا می‌کند؟



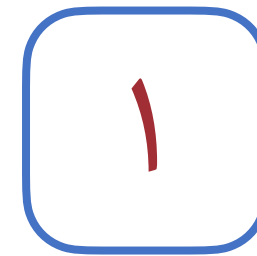
باقیمانده‌ی پول مشتری



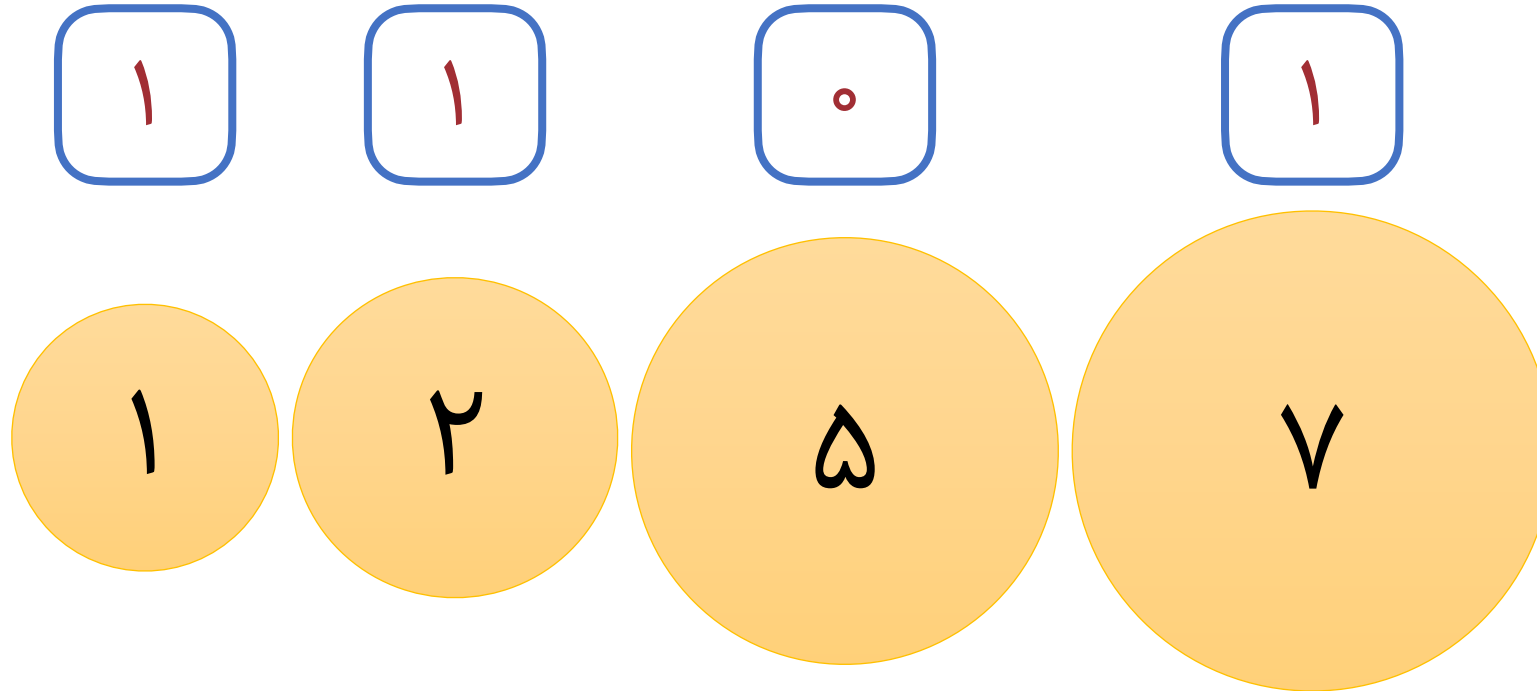
آیا الگوریتم حریمانه جواب بهینه را پیدا می‌کند؟



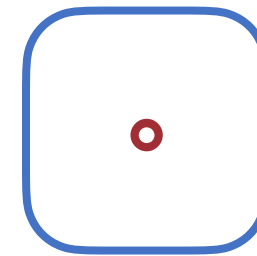
باقیمانده‌ی پول مشتری



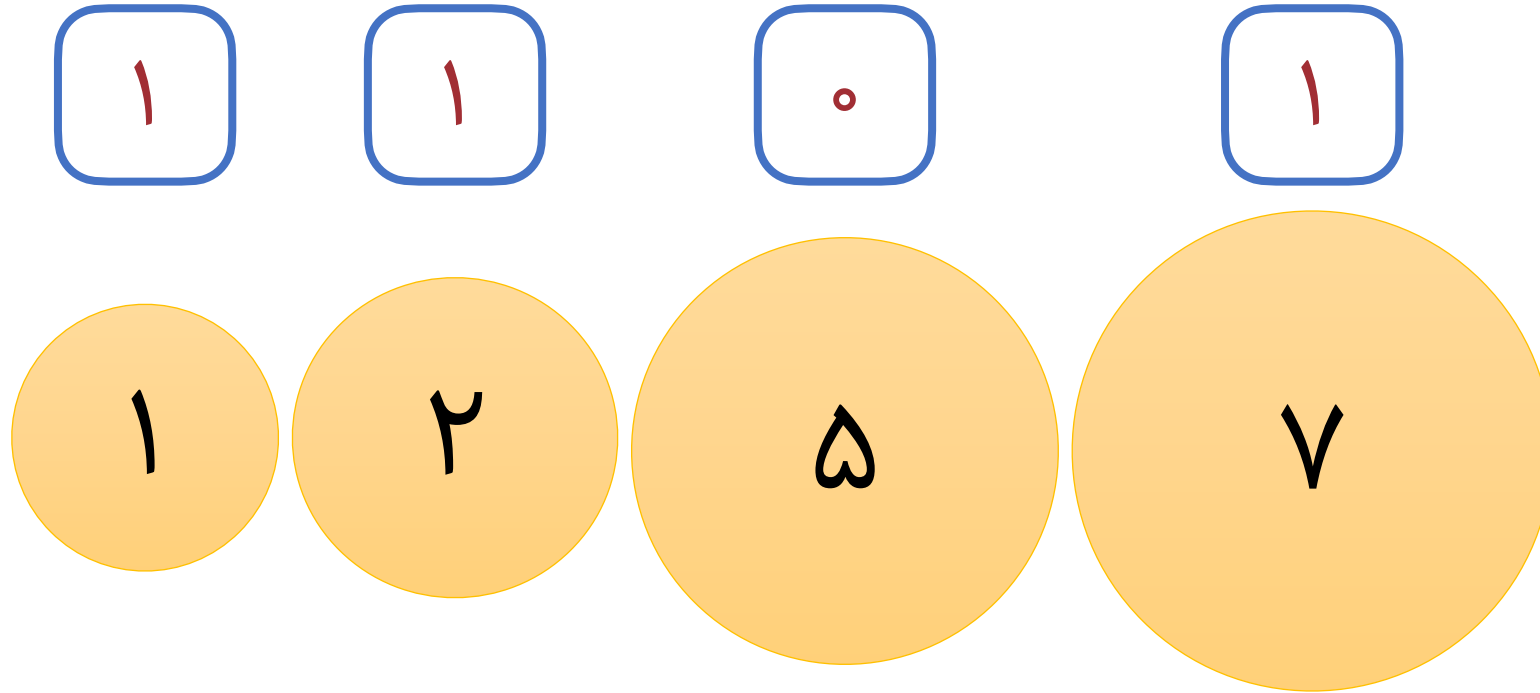
آیا الگوریتم حریمانه جواب بهینه را پیدا می‌کند؟



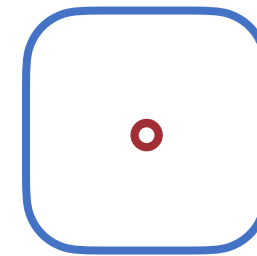
باقیمانده‌ی پول مشتری



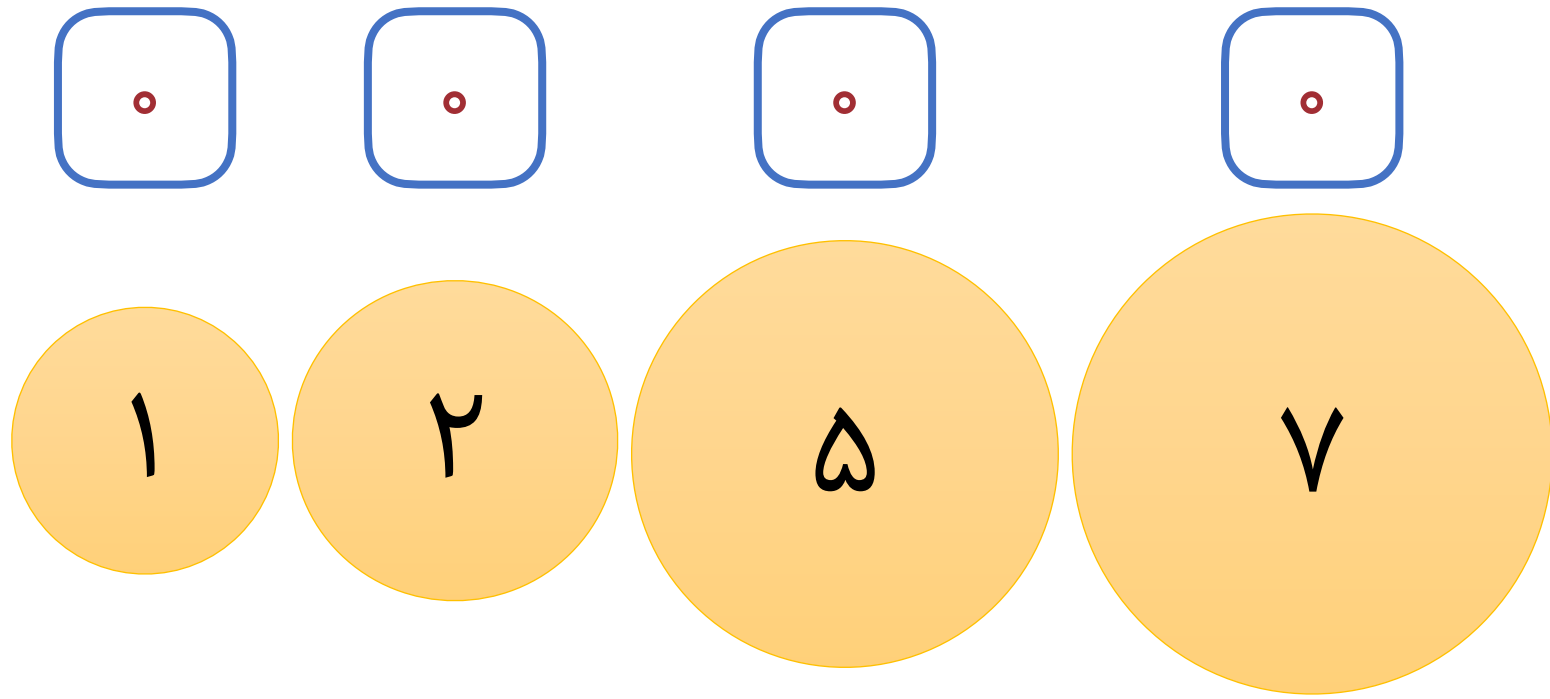
۳ سکه استفاده کردیم.



باقیمانده‌ی پول مشتری



ولی پاسخ درست ۲ سکه است.

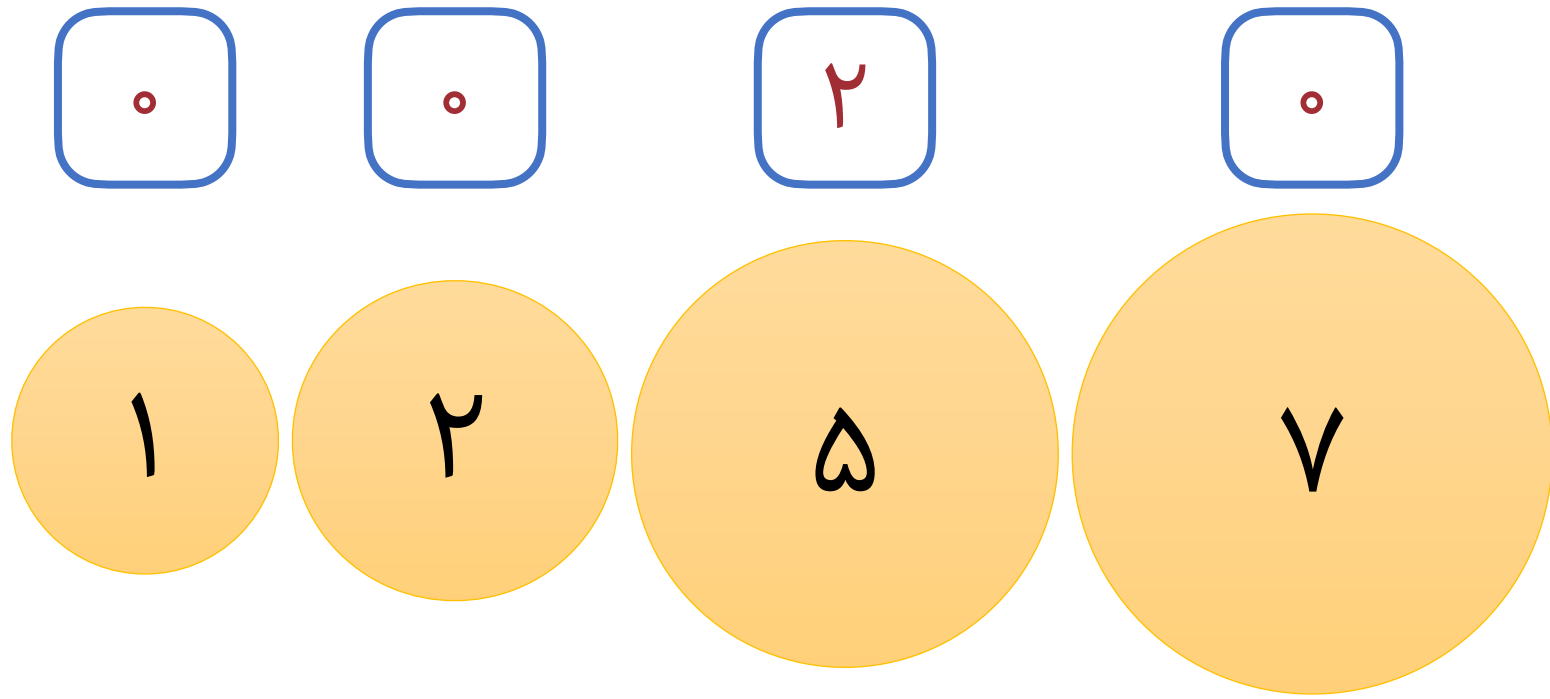


باقیمانده‌ی پول مشتری

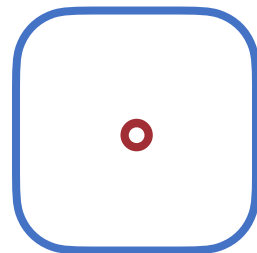
۱۰



ولی پاسخ درست ۲ سکه است.

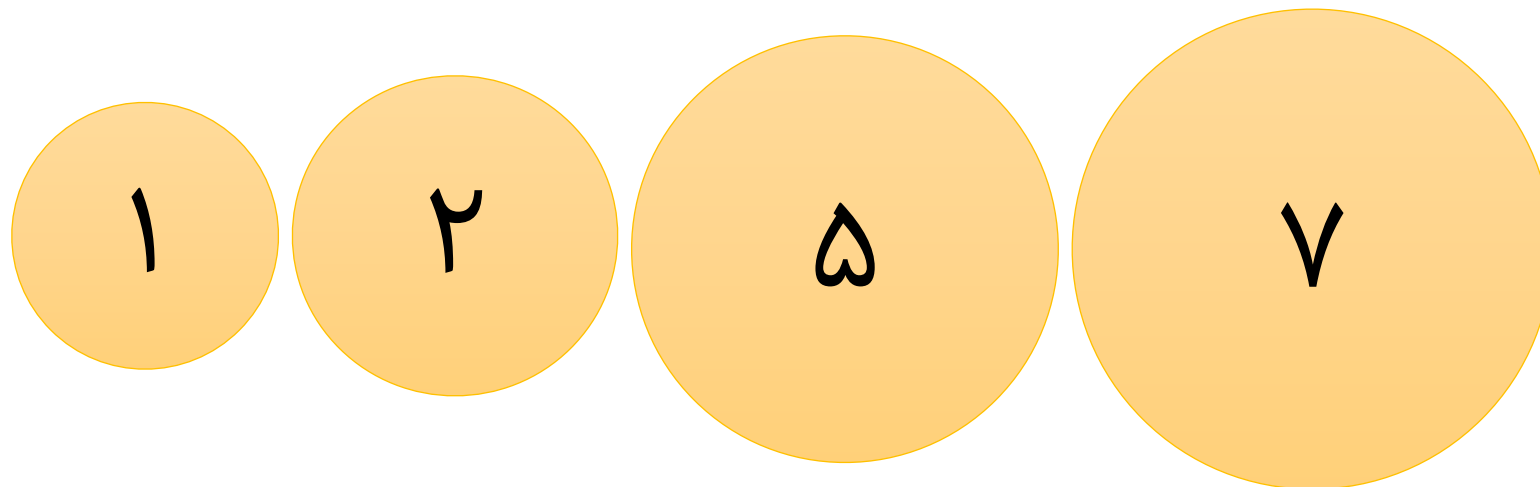


باقیمانده‌ی پول مشتری

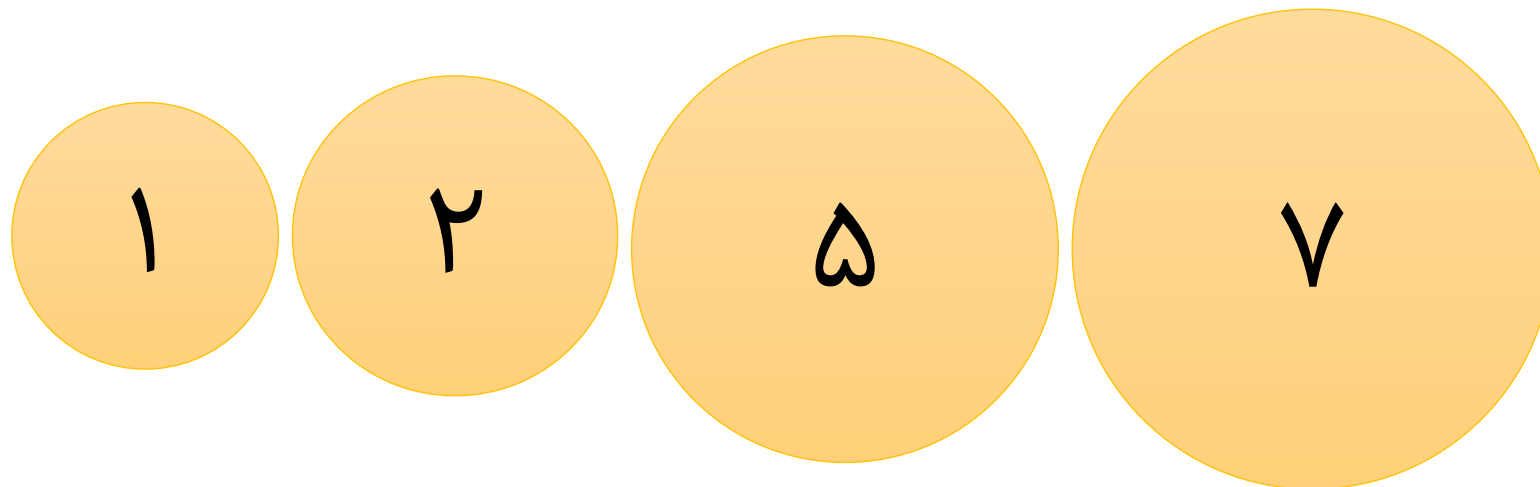


الگوریتم حریرصانه برای بعضی ترکیب سکه‌ها بهترین جواب را پیدا نمی‌کند.

بیابید مساله را از ابتدا حل کنیم.

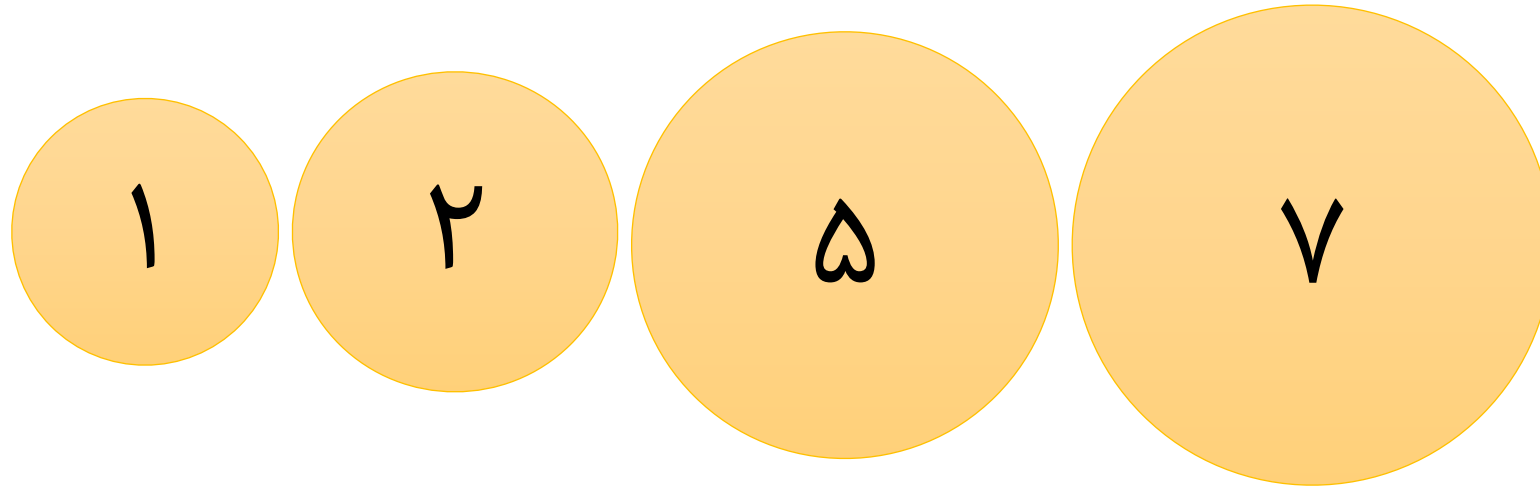


بیاپید مساله را از ابتدا حل کنیم.



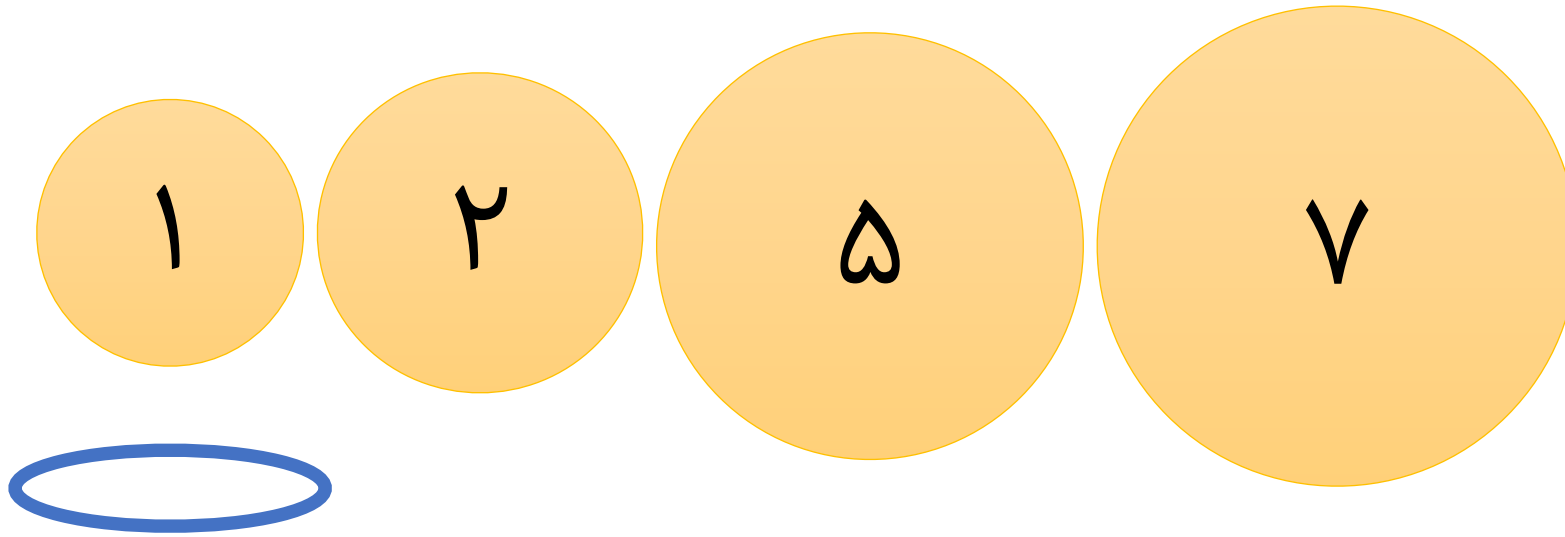
										♦	پول
										♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



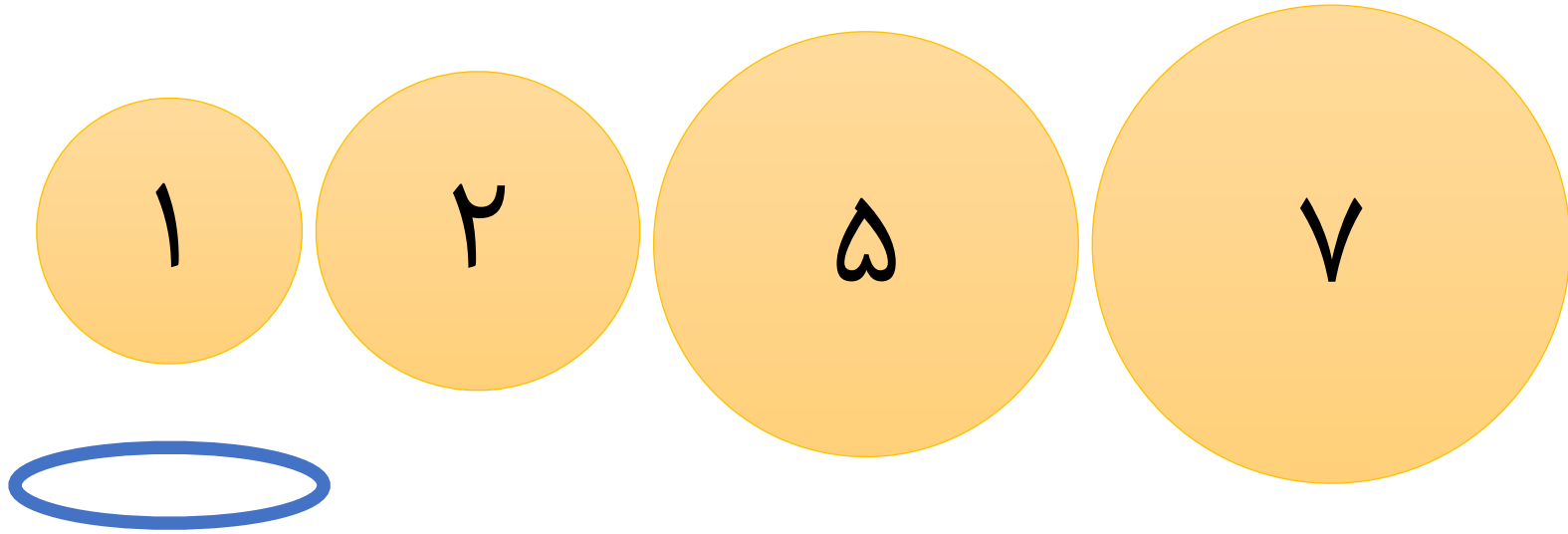
										۱	♦	پول
											♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



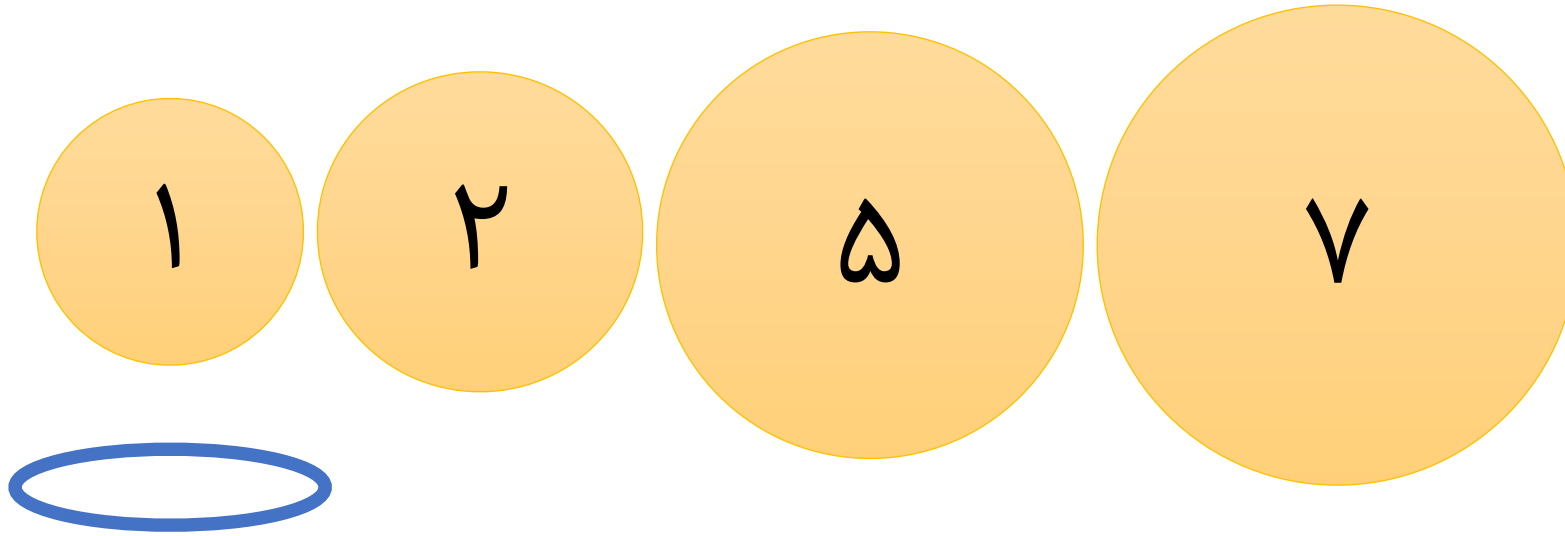
										۱	♦	پول
											♦	تعداد سکه

بیاپید مساله را از ابتدا حل کنیم.



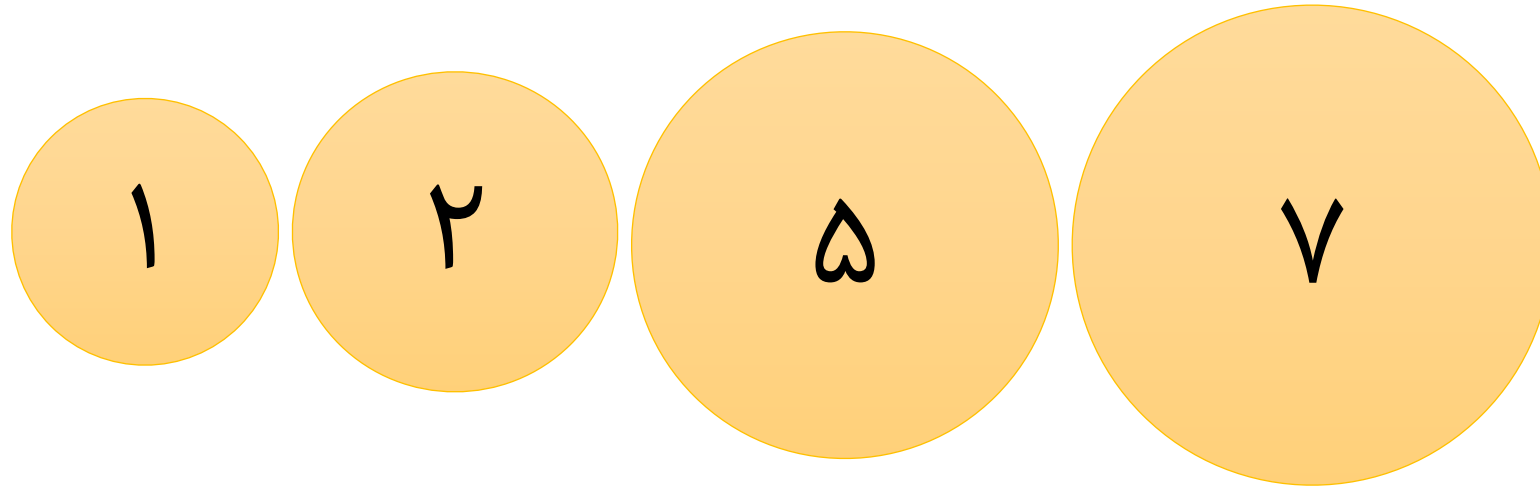
										۱	♦	پول
											♦	تعداد سکه

بیاید مساله را از ابتدا حل کنیم.



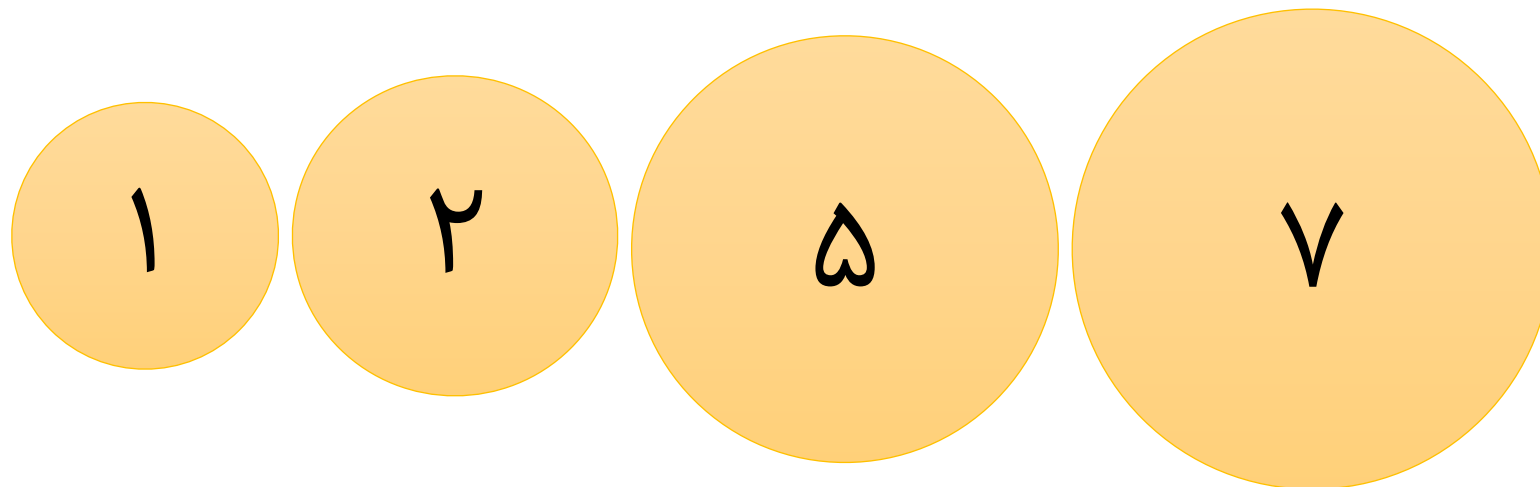
										۱	♦	پول
										۱	♦	تعداد سکه

بیاید مساله را از ابتدا حل کنیم.



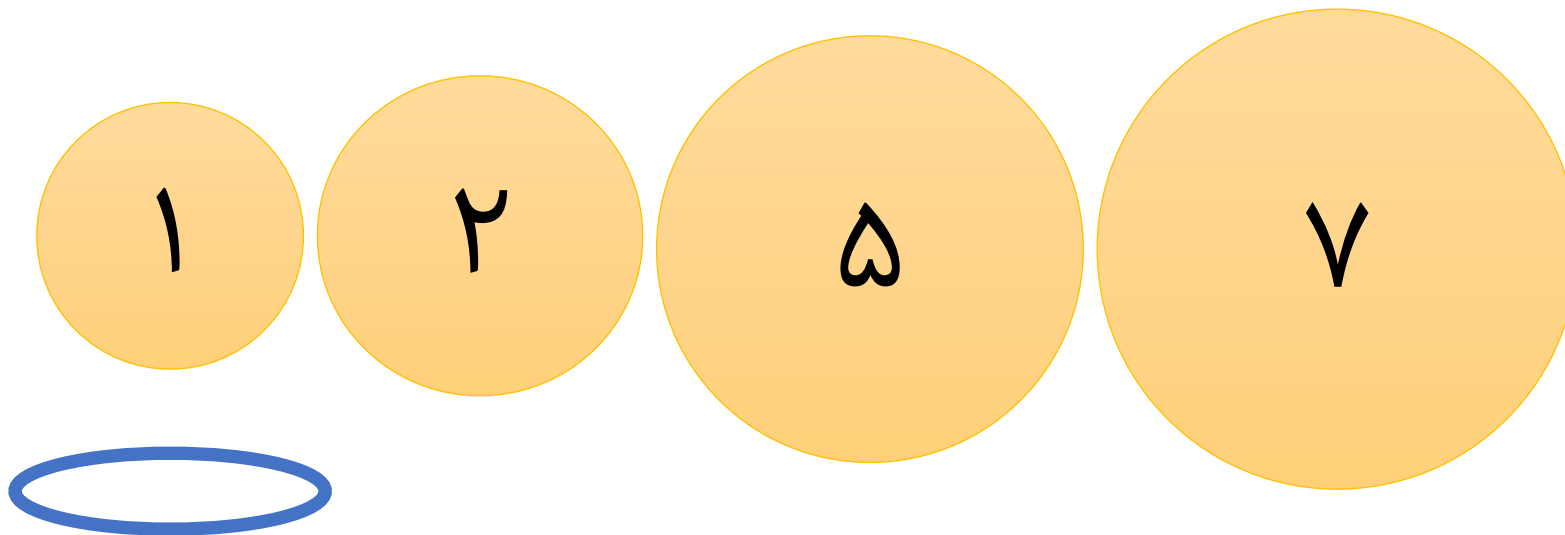
										۱	♦	پول
										۱	♦	تعداد سکه

بیاید مساله را از ابتدا حل کنیم.



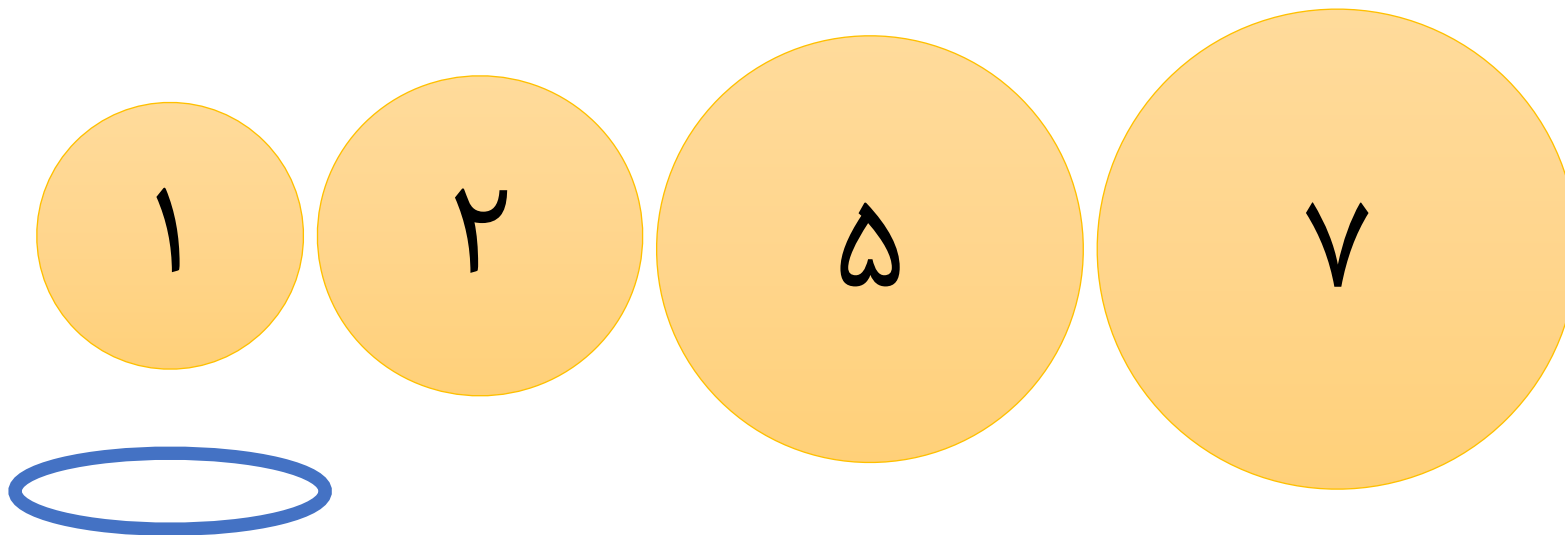
								۲	۱	♦	پول
									۱	♦	تعداد سکه

بیاید مساله را از ابتدا حل کنیم.



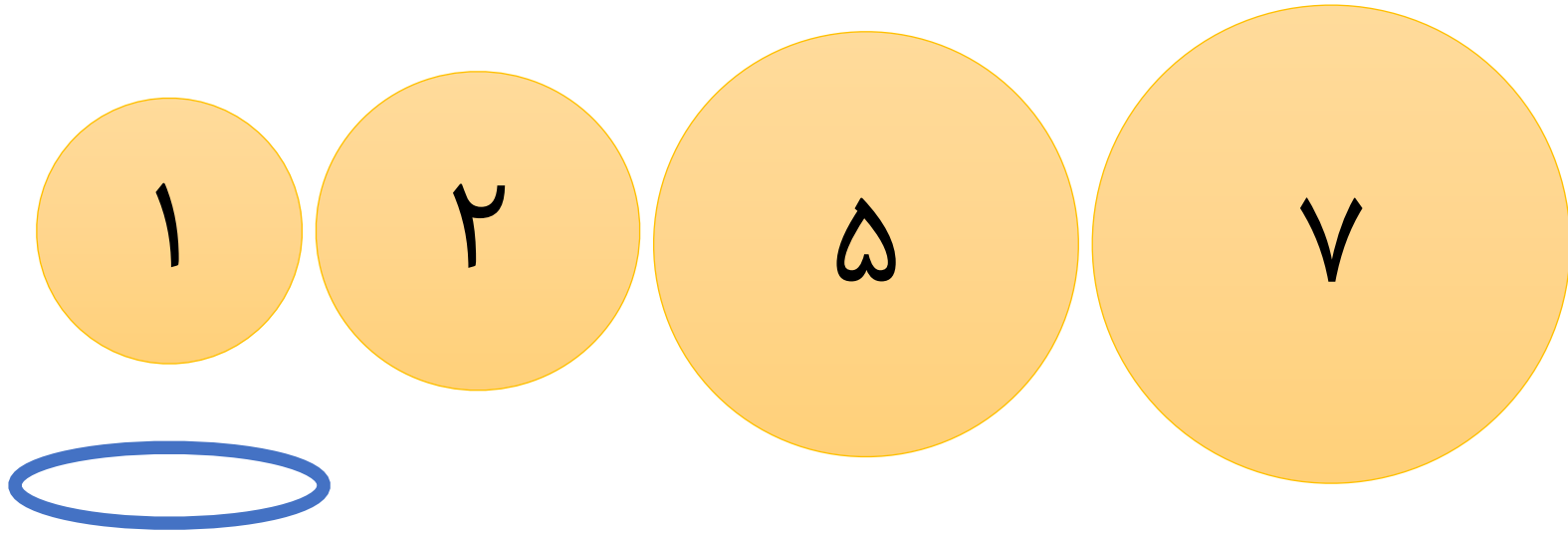
								۲	۱	♦	پول
									۱	♦	تعداد سکه

بیاپید مساله را از ابتدا حل کنیم.



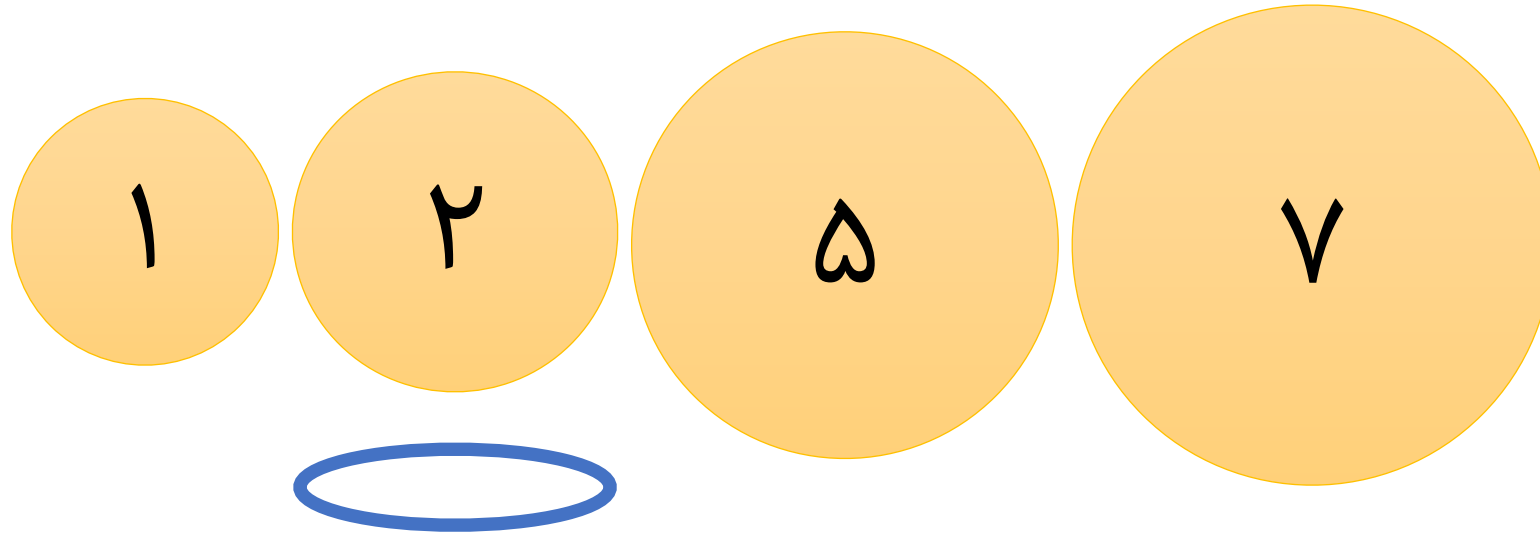
								۲	۱	♦	پول
									۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



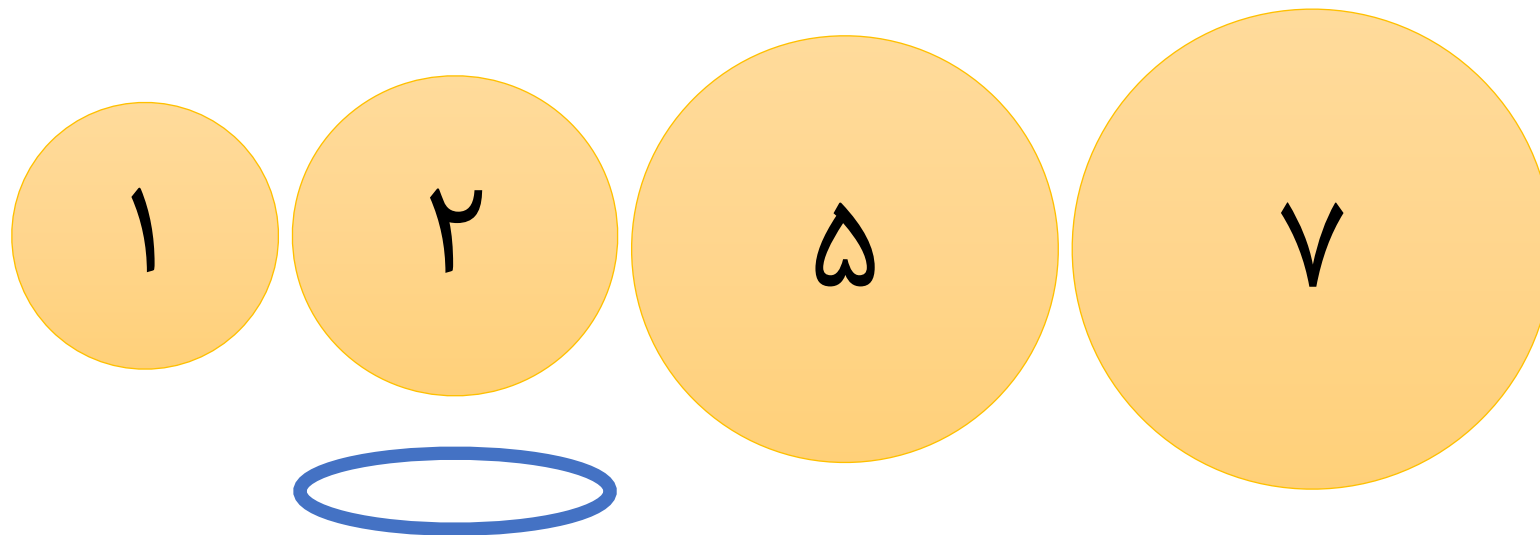
								۲	۱	♦	پول
								۲	۱	♦	تعداد سکه

بیابید مساله را از ابتدا حل کنیم.



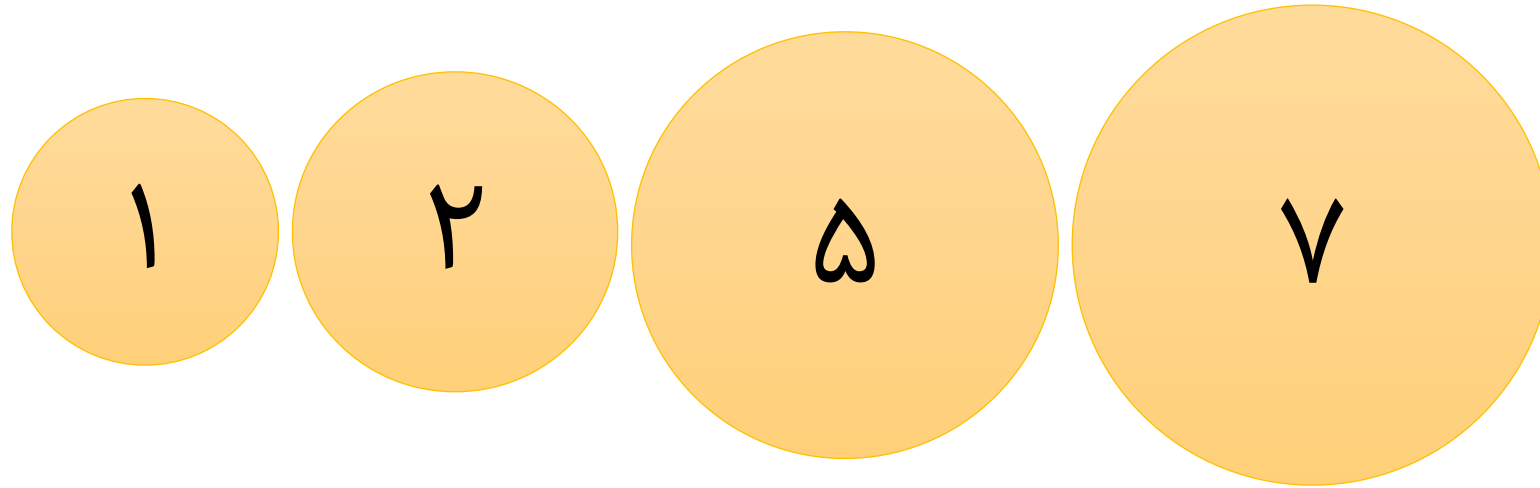
								۲	۱	♦	پول
								۲	۱	♦	تعداد سکه

بیاپید مساله را از ابتدا حل کنیم.



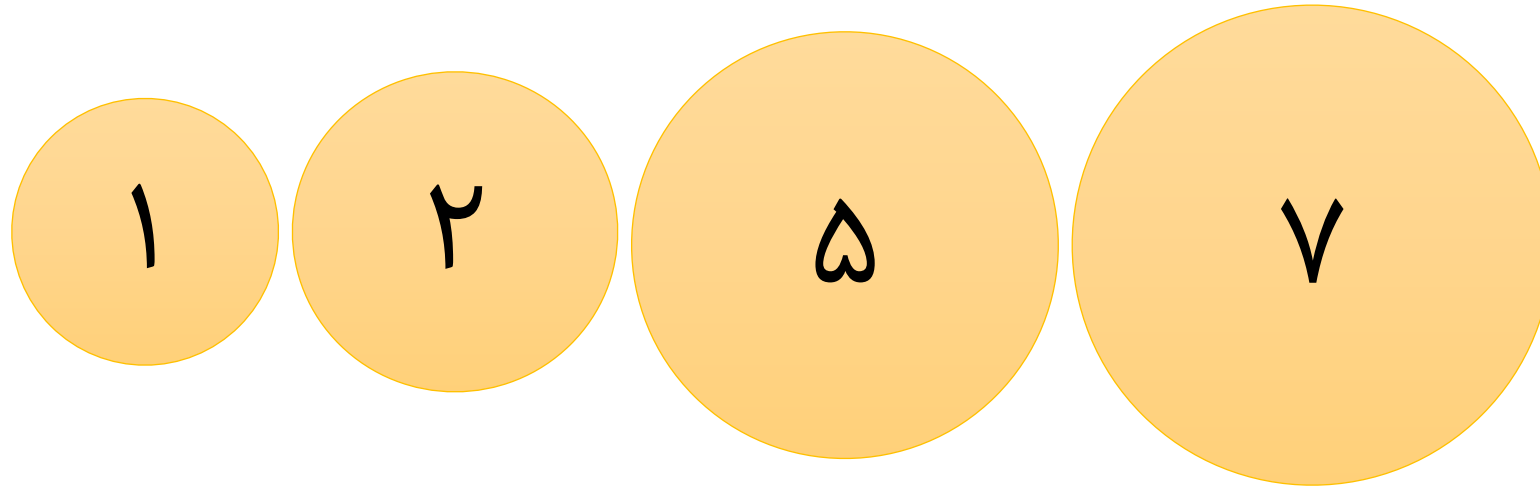
									۲	۱	♦	پول
									۱	۱	♦	تعداد سکه

بیاید مساله را از ابتدا حل کنیم.



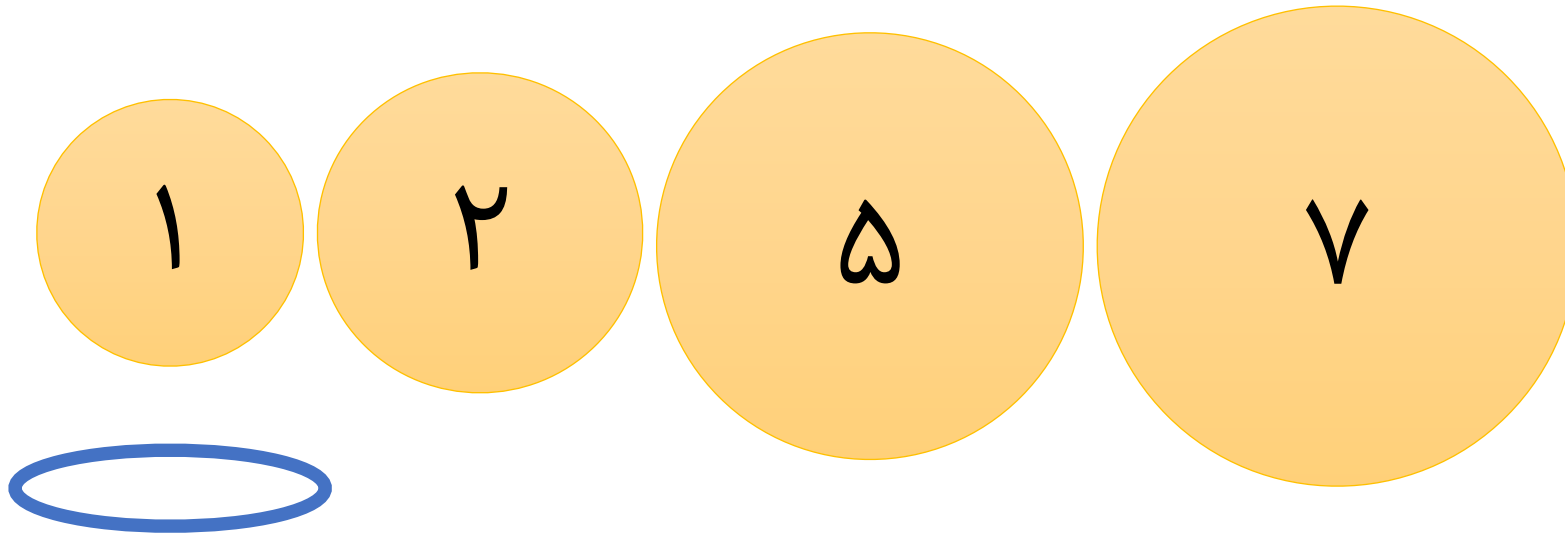
								۲	۱	♦	پول
								۱	۱	♦	تعداد سکه

بیاپید مساله را از ابتدا حل کنیم.



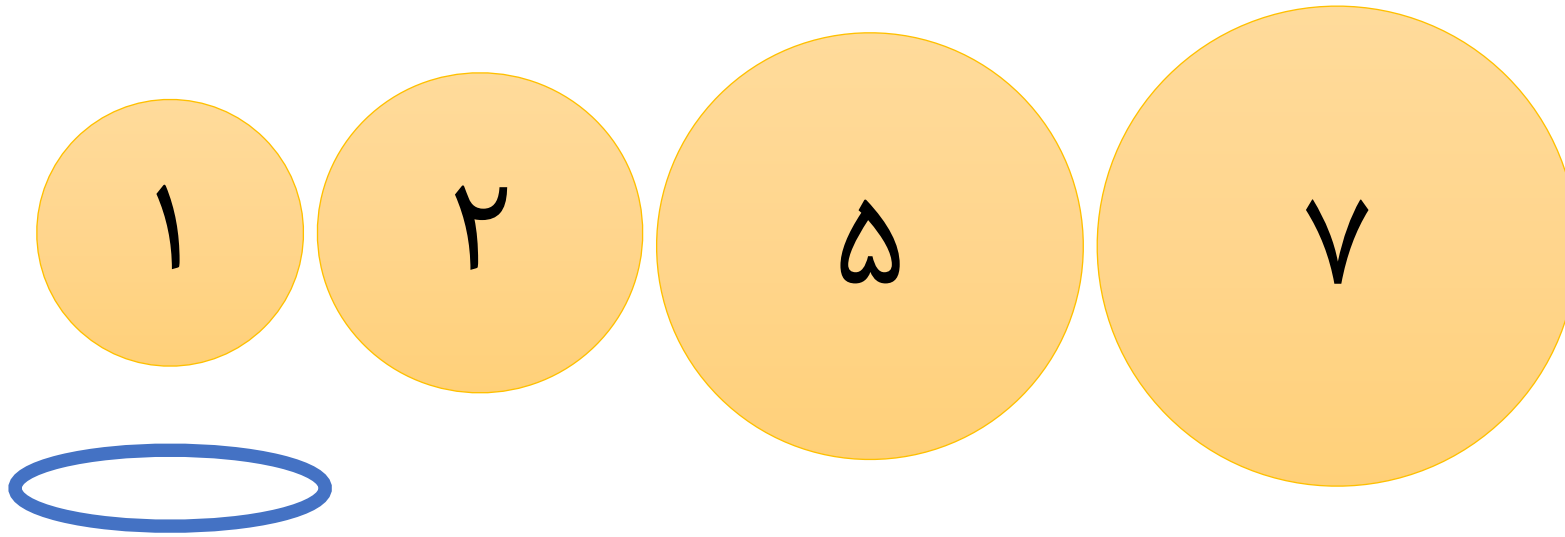
							۳	۲	۱	♦	پول
								۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



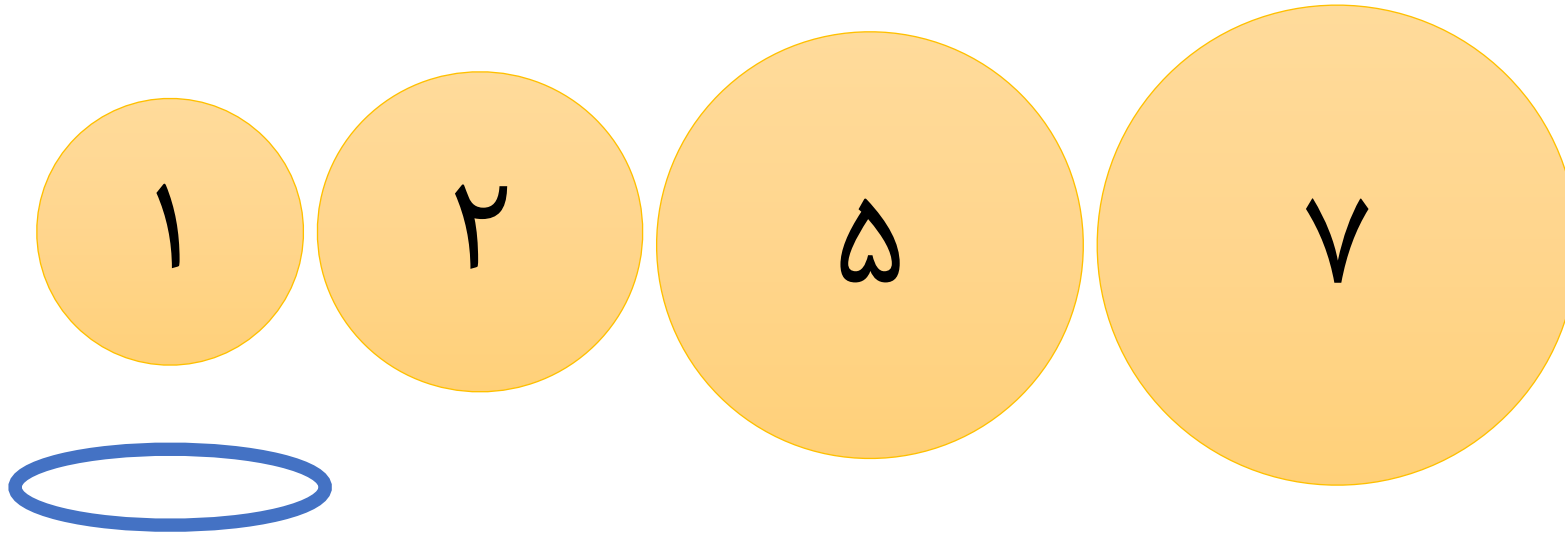
							۳	۲	۱	♦	پول
								۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



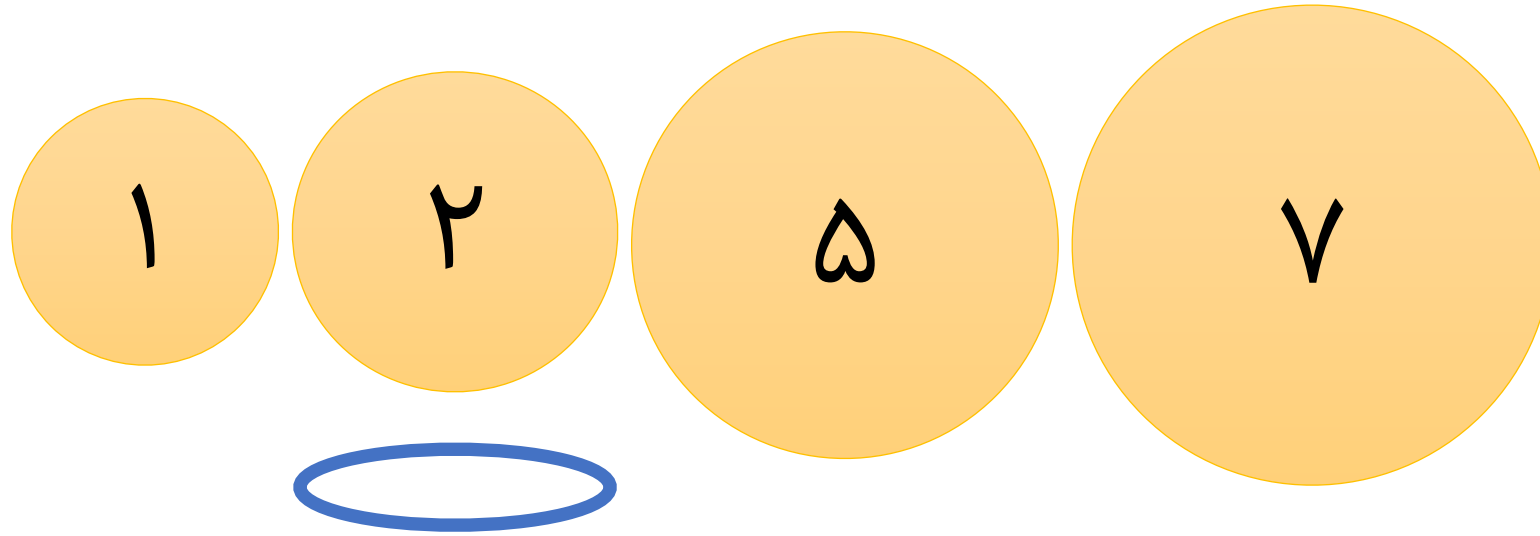
							۳	۲	۱	♦	پول
								۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



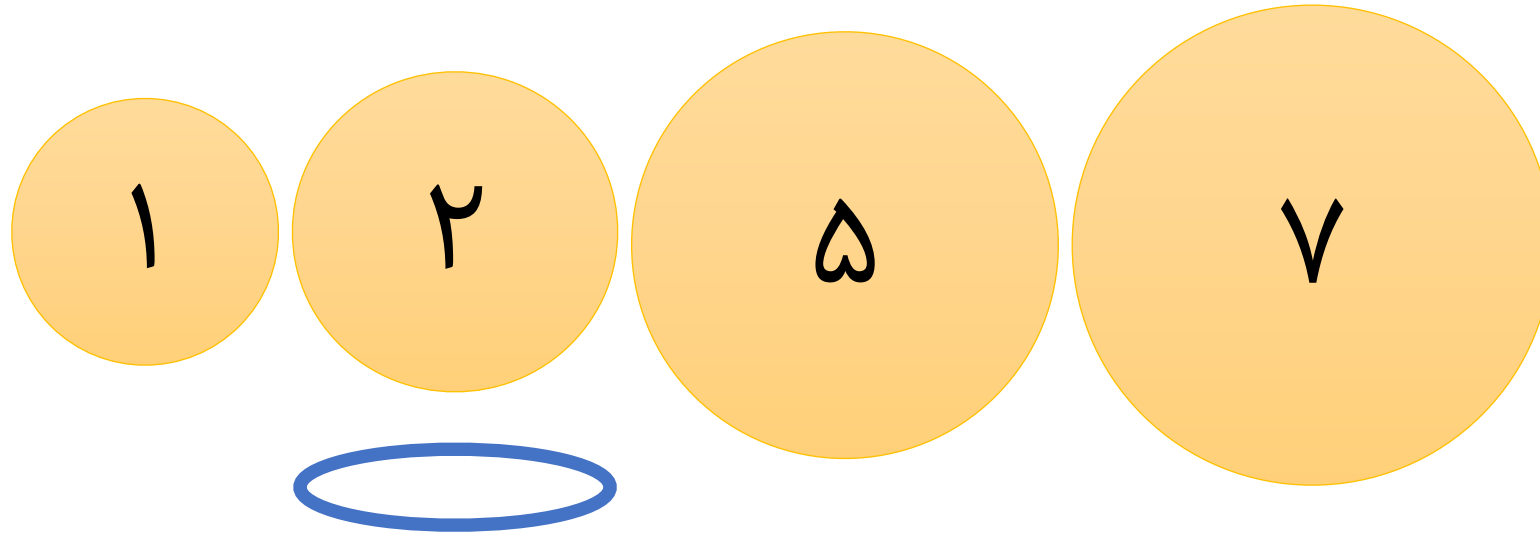
							۳	۲	۱	♦	پول
							۲	۱	۱	♦	تعداد سکه

بیاید مساله را از ابتدا حل کنیم.



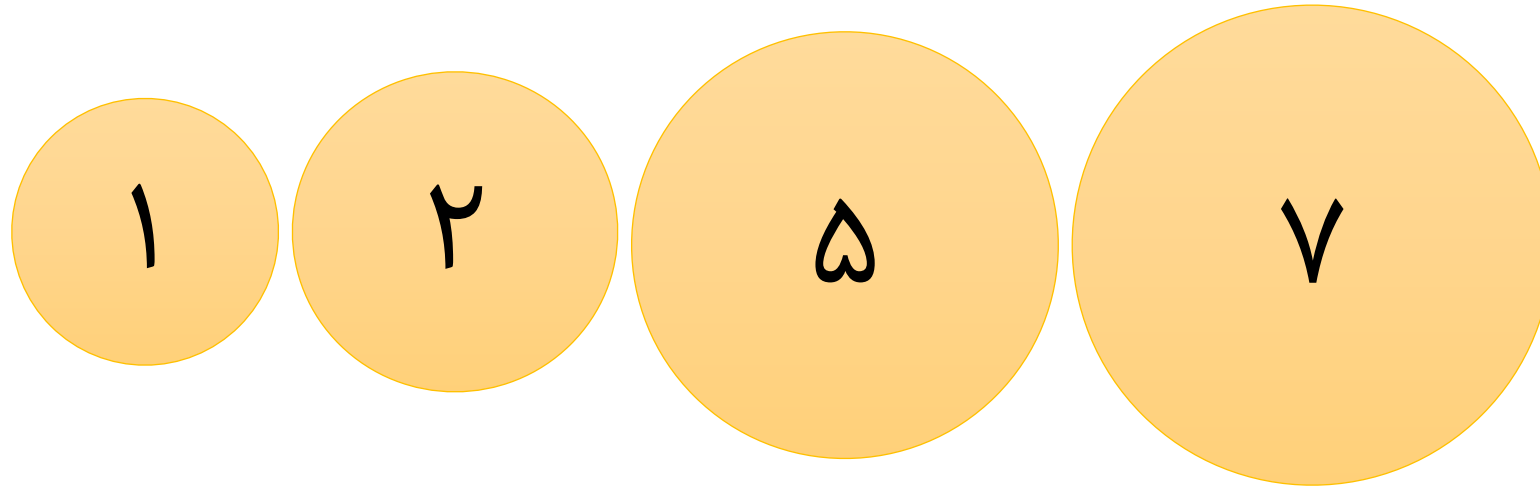
							۳	۲	۱	♦	پول
							۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



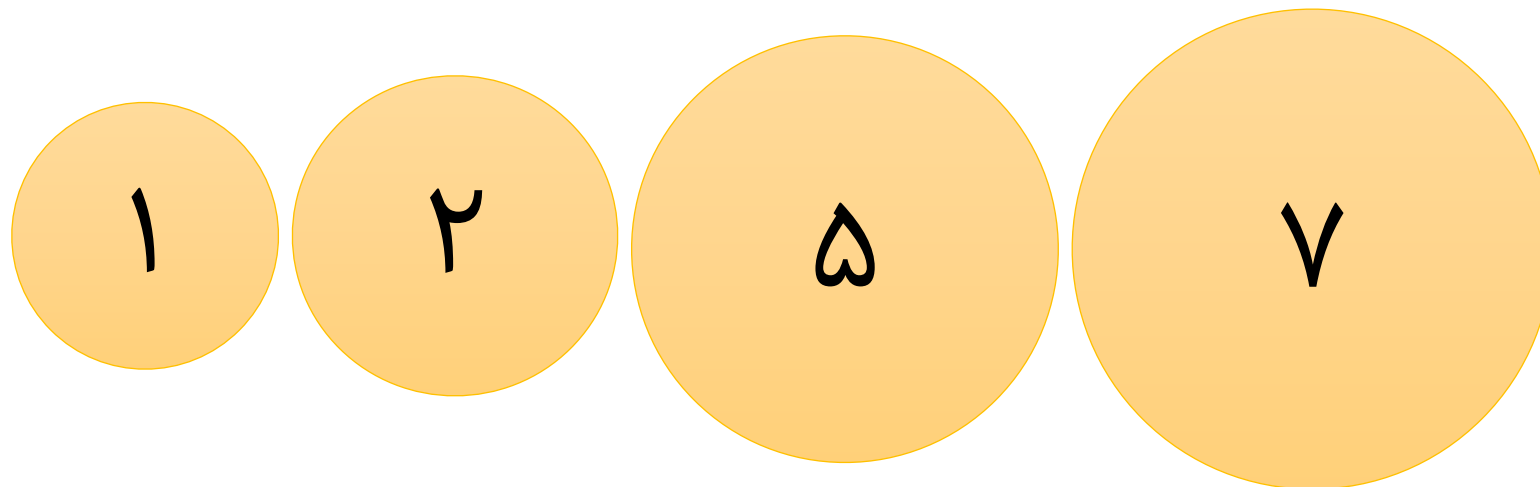
							۳	۲	۱	♦	پول
							۲	۱	۱	♦	تعداد سکه

بیاید مساله را از ابتدا حل کنیم.



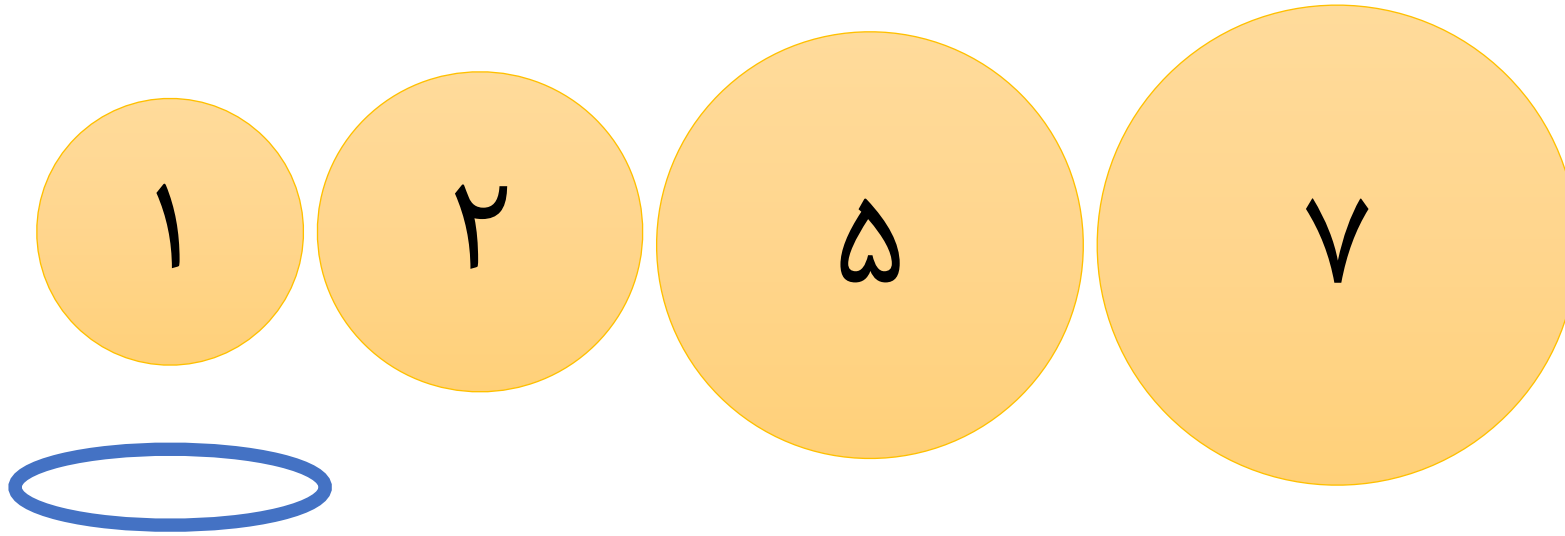
							۳	۲	۱	♦	پول
							۲	۱	۱	♦	تعداد سکه

بیاپید مساله را از ابتدا حل کنیم.



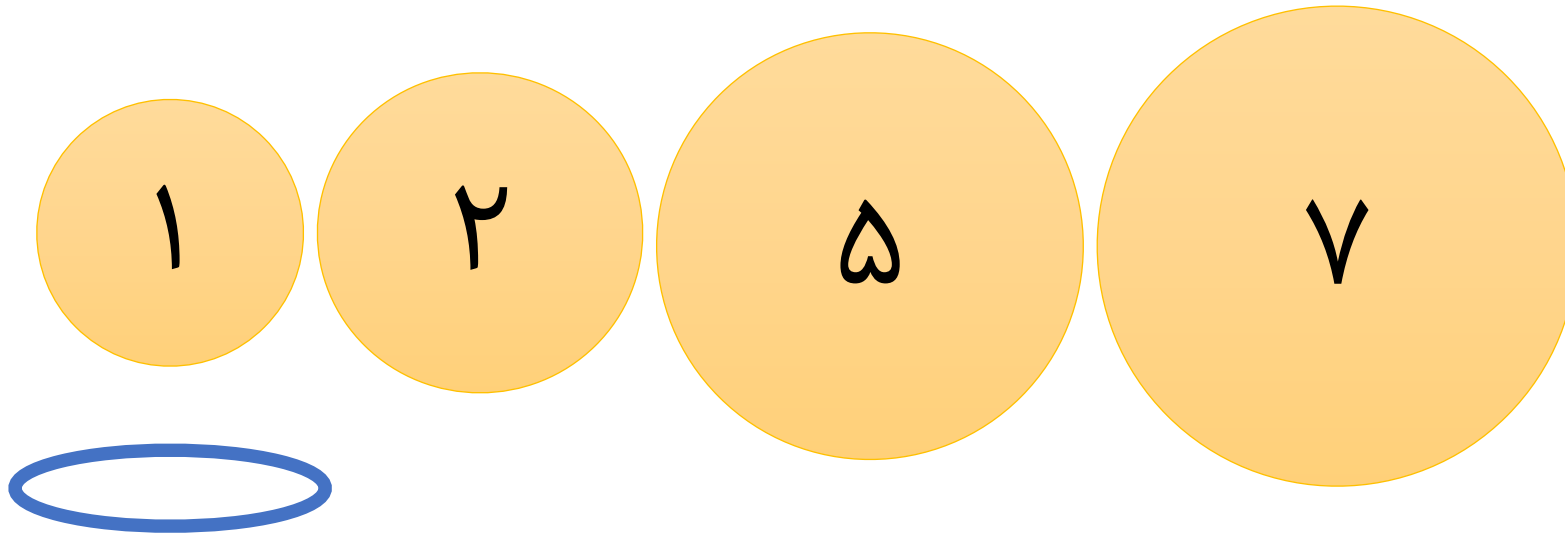
						۴	۳	۲	۱	♦	پول
							۲	۱	۱	♦	تعداد سکه

بیاپید مساله را از ابتدا حل کنیم.



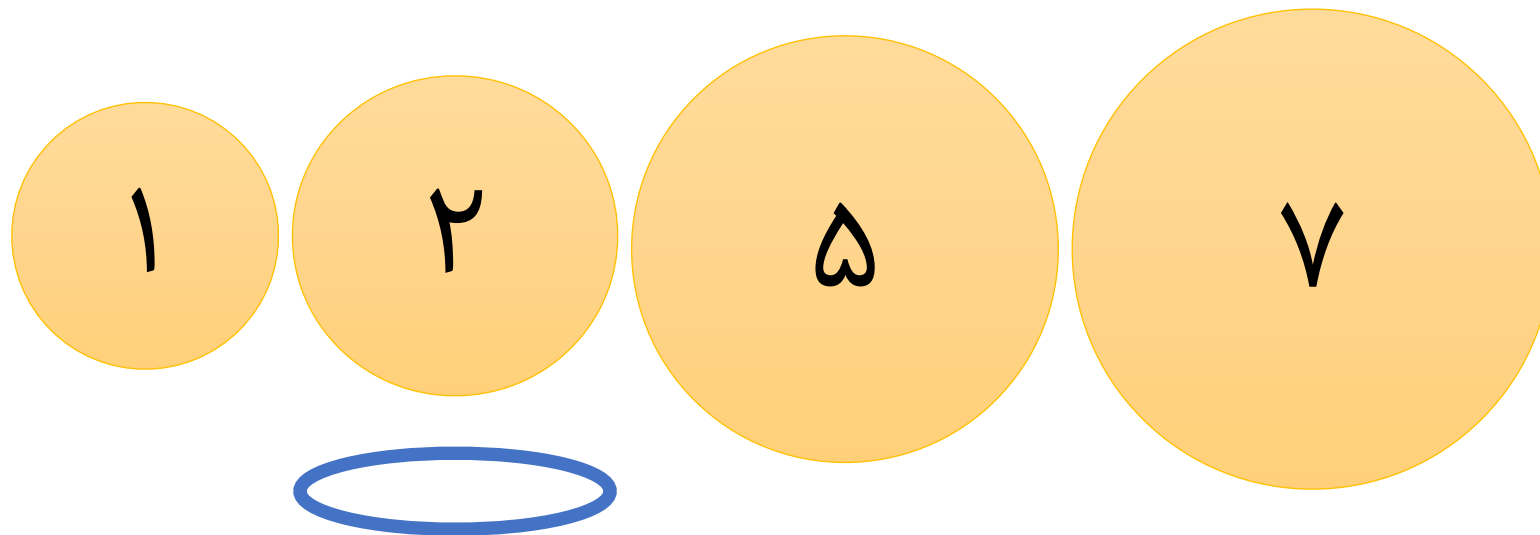
						۴	۳	۲	۱	♦	پول
							۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



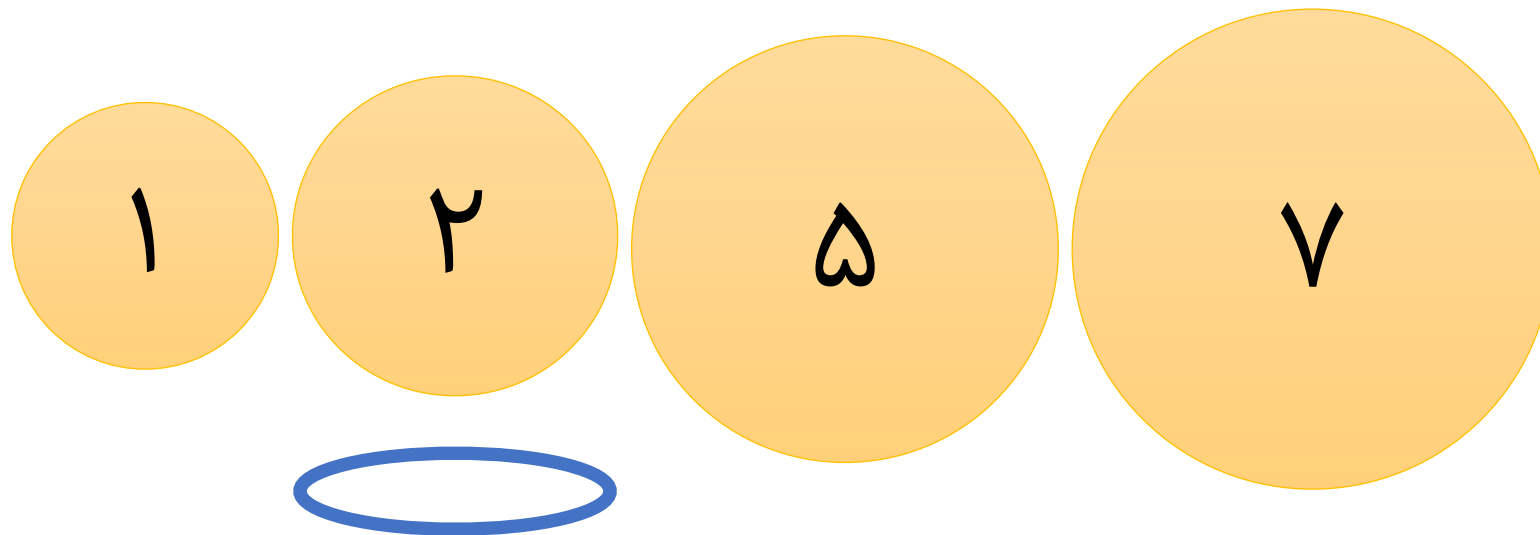
						۴	۳	۲	۱	♦	پول
						۳	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



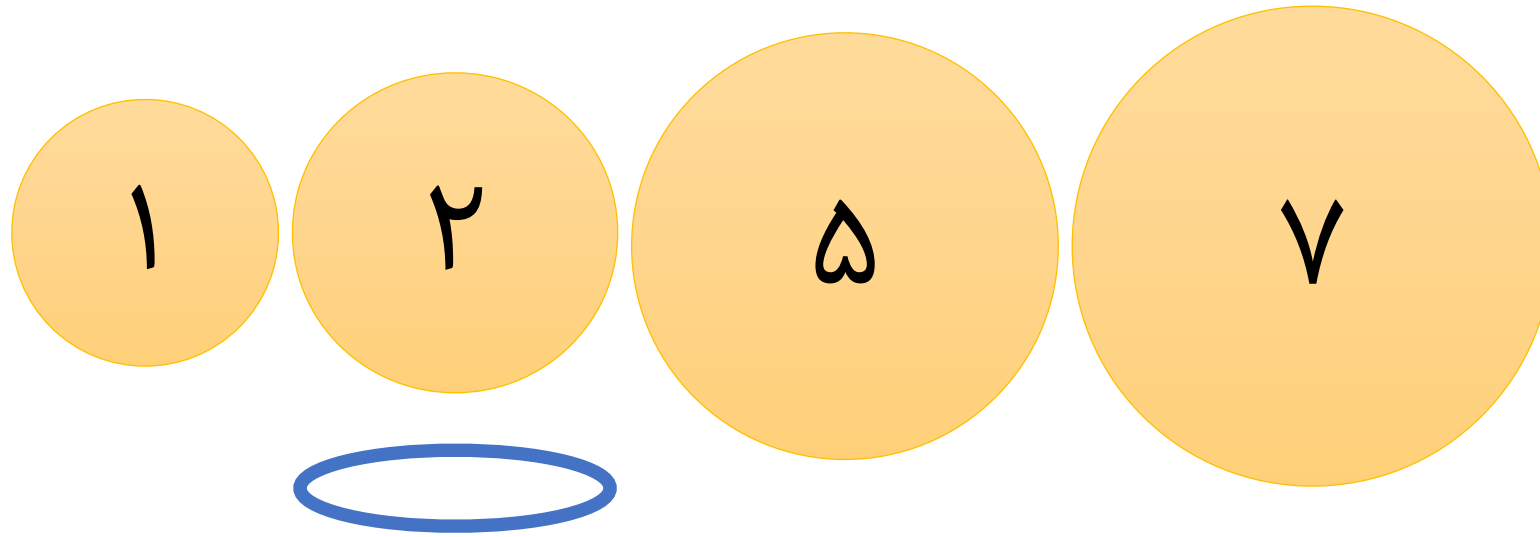
						۴	۳	۲	۱	♦	پول
						۳	۲	۱	۱	♦	تعداد سکه

بیاپید مساله را از ابتدا حل کنیم.



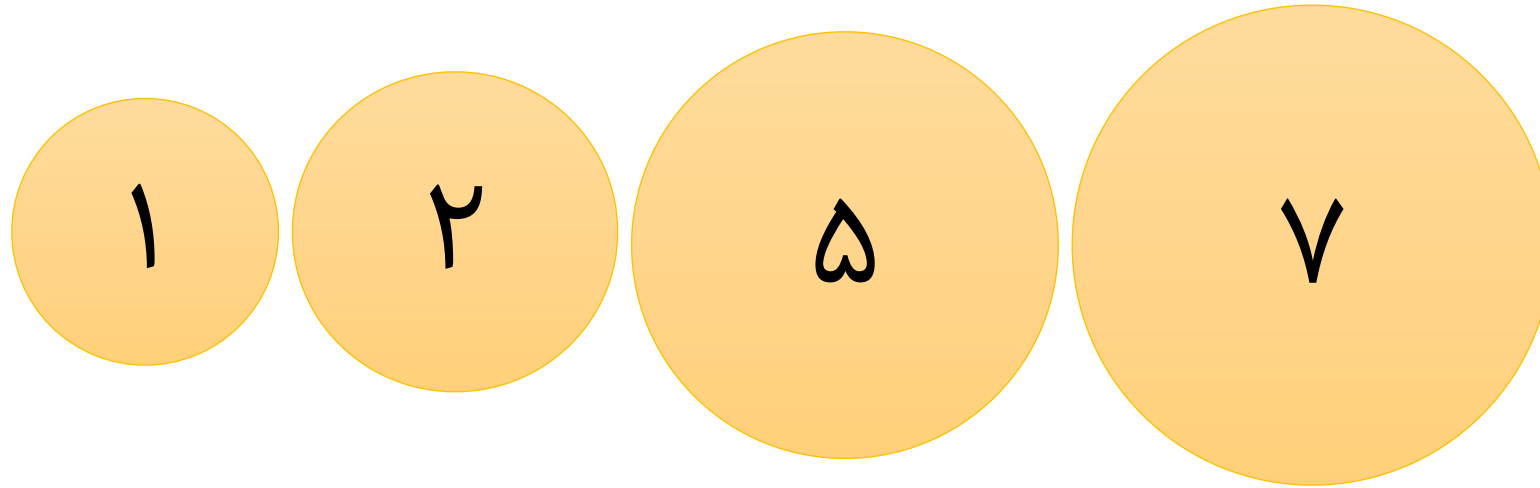
						۴	۳	۲	۱	♦	پول
						۳	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



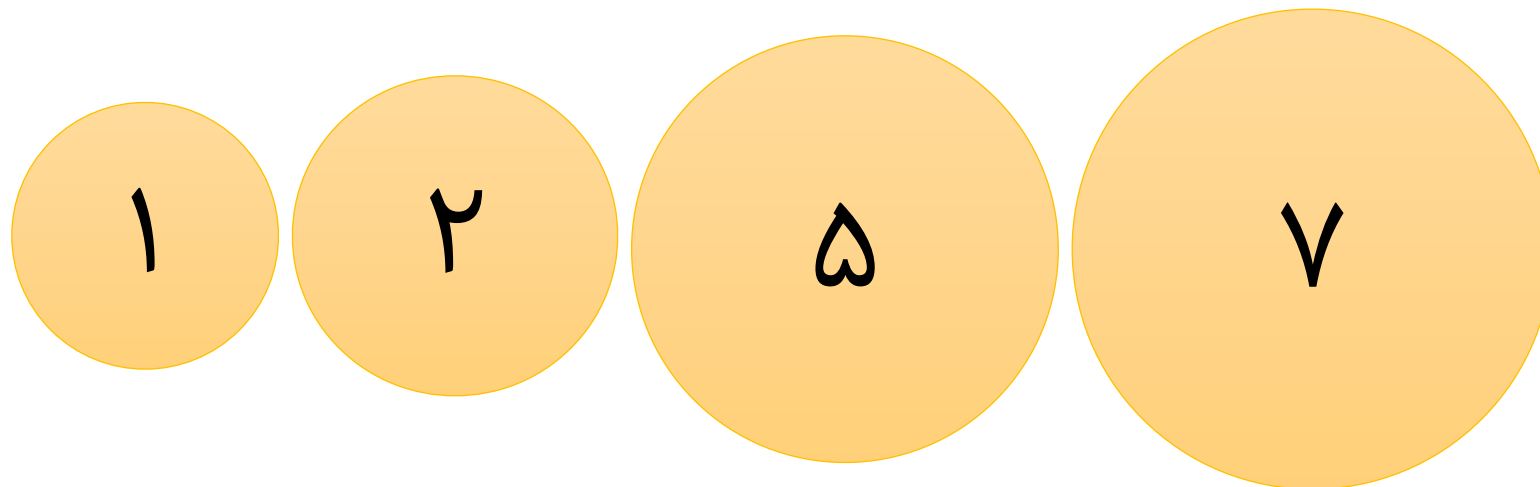
						۴	۳	۲	۱	♦	پول
						۲	۲	۱	۱	♦	تعداد سکه

بیاید مساله را از ابتدا حل کنیم.



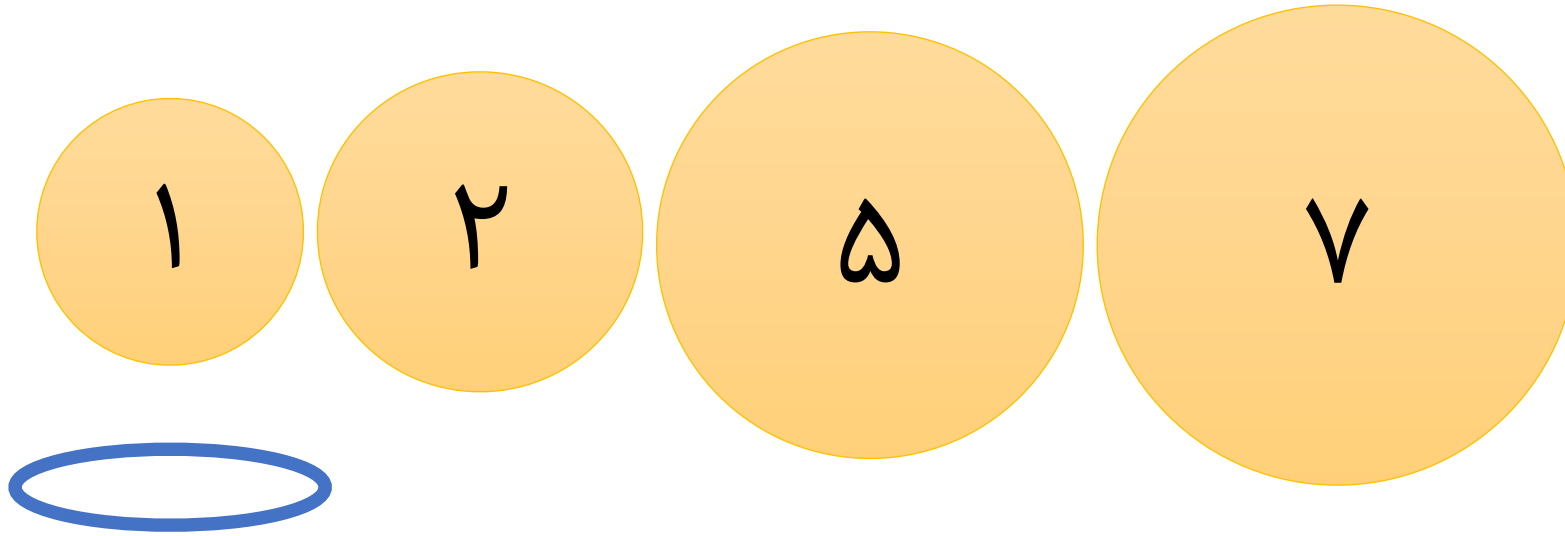
						۴	۳	۲	۱	♦	پول
						۲	۲	۱	۱	♦	تعداد سکه

بیاپید مساله را از ابتدا حل کنیم.



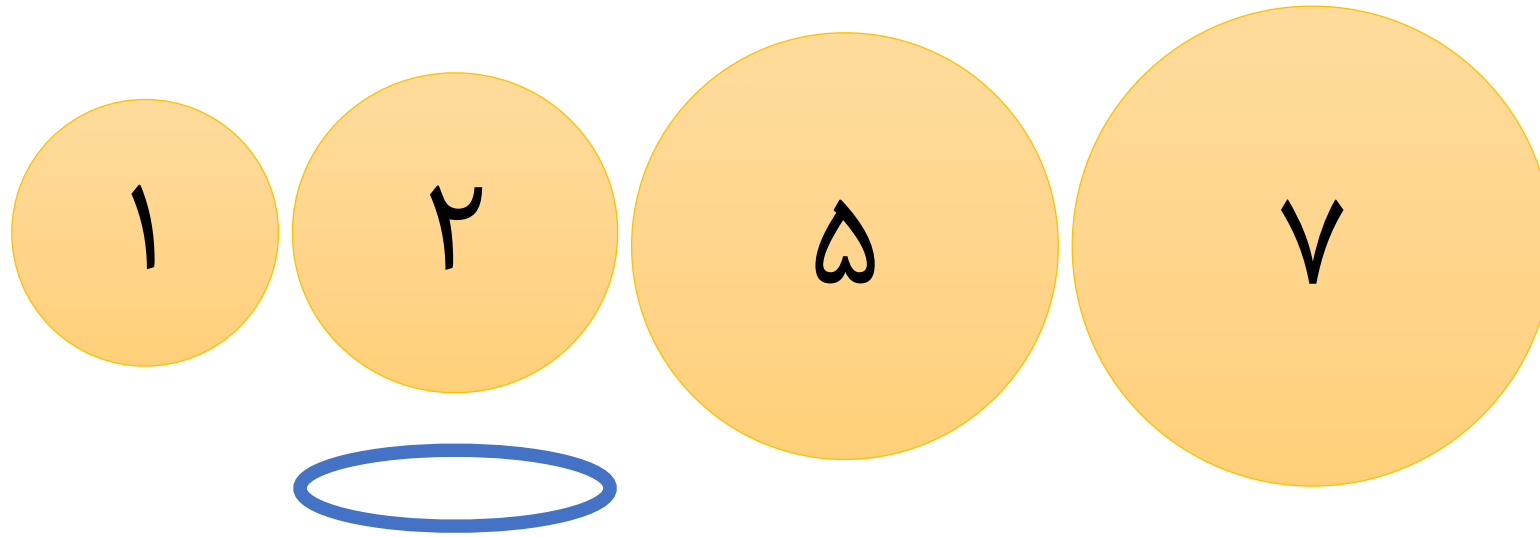
					۵	۴	۳	۲	۱	♦	پول
						۲	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



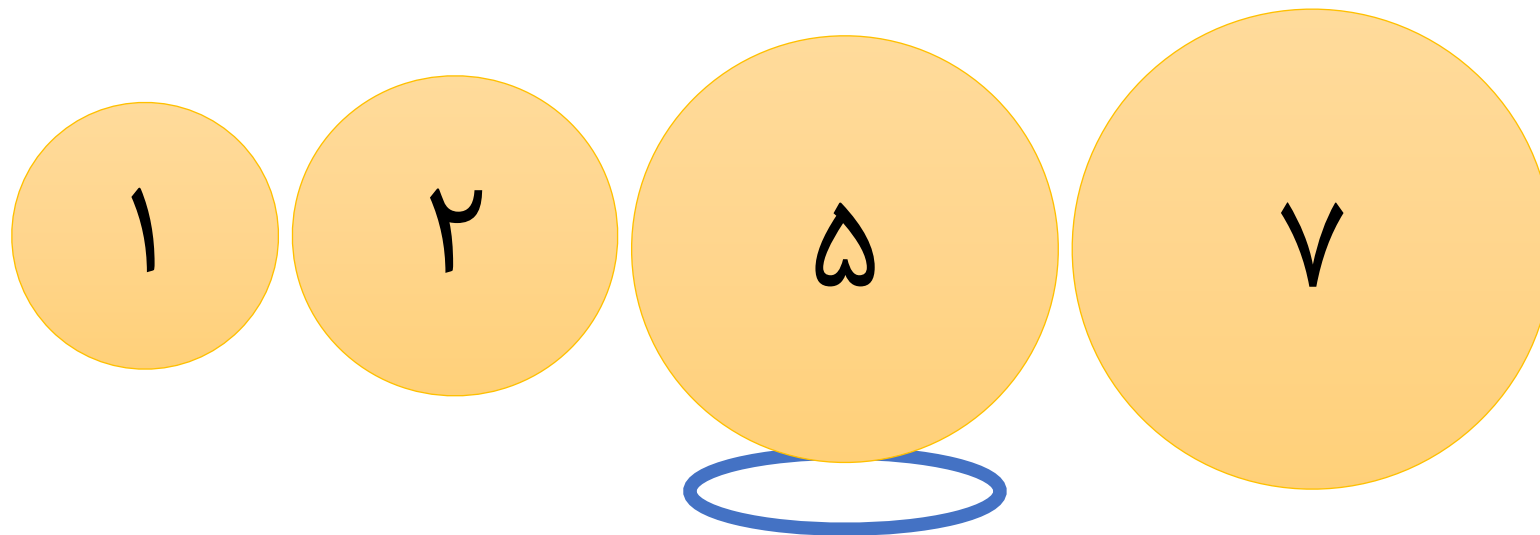
					۵	۴	۳	۲	۱	♦	پول
					۳	۲	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



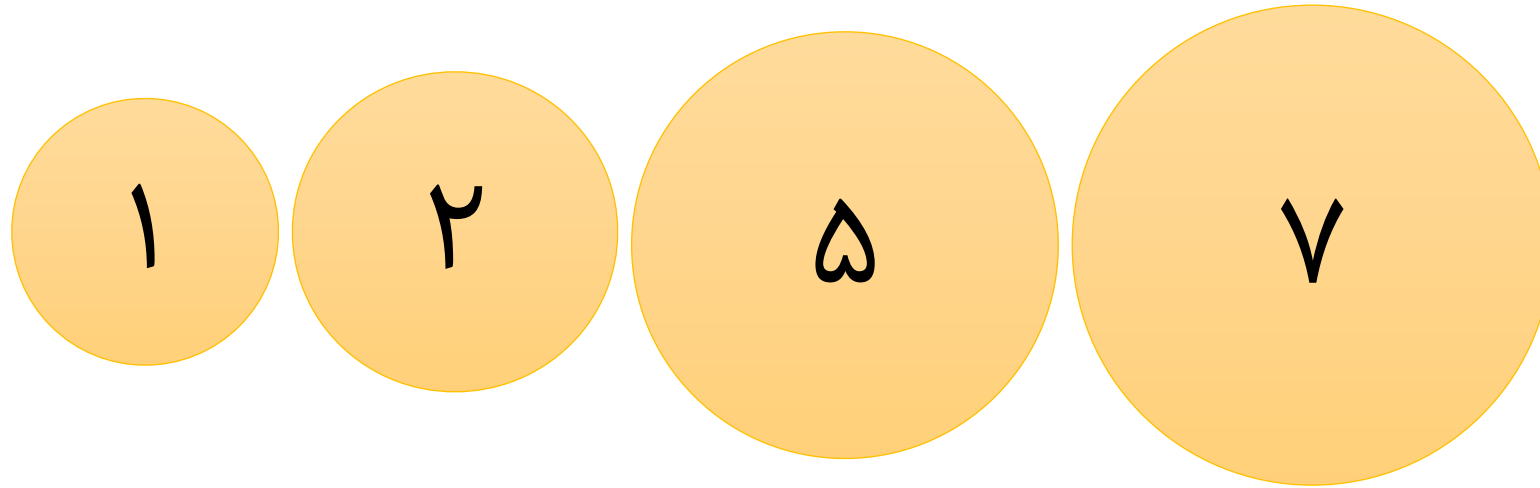
					۵	۴	۳	۲	۱	♦	پول
					۳	۲	۲	۱	۱	♦	تعداد سکه

بیاپید مساله را از ابتدا حل کنیم.



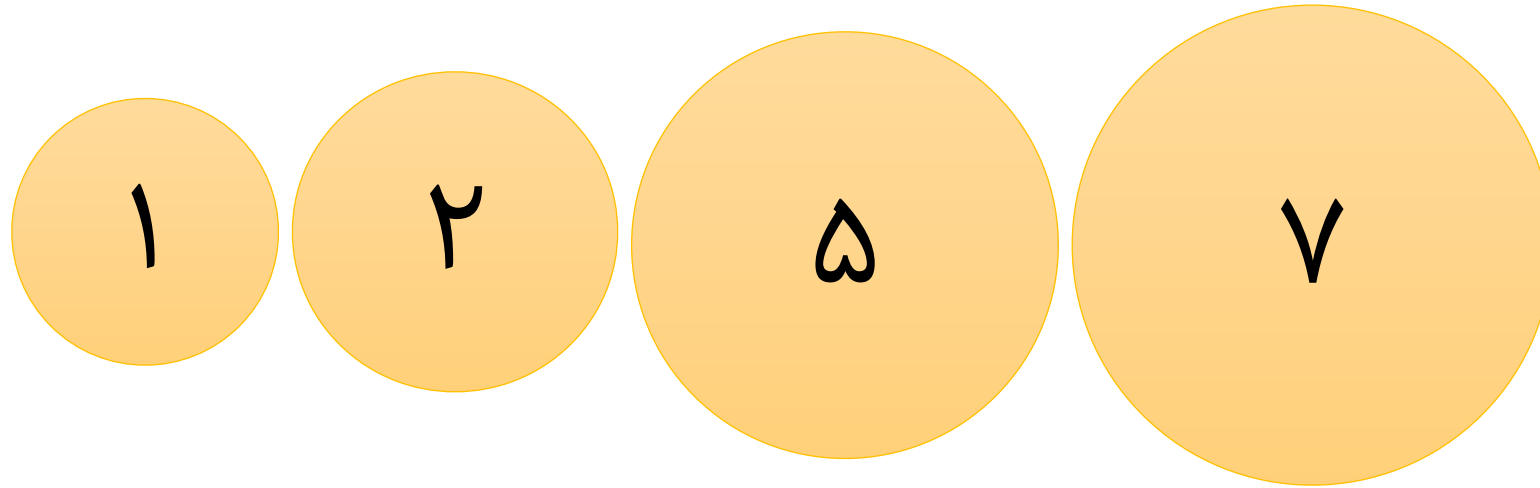
					۵	۴	۳	۲	۱	♦	پول
					۱	۲	۲	۱	۱	♦	تعداد سکه

بیاپید مساله را از ابتدا حل کنیم.



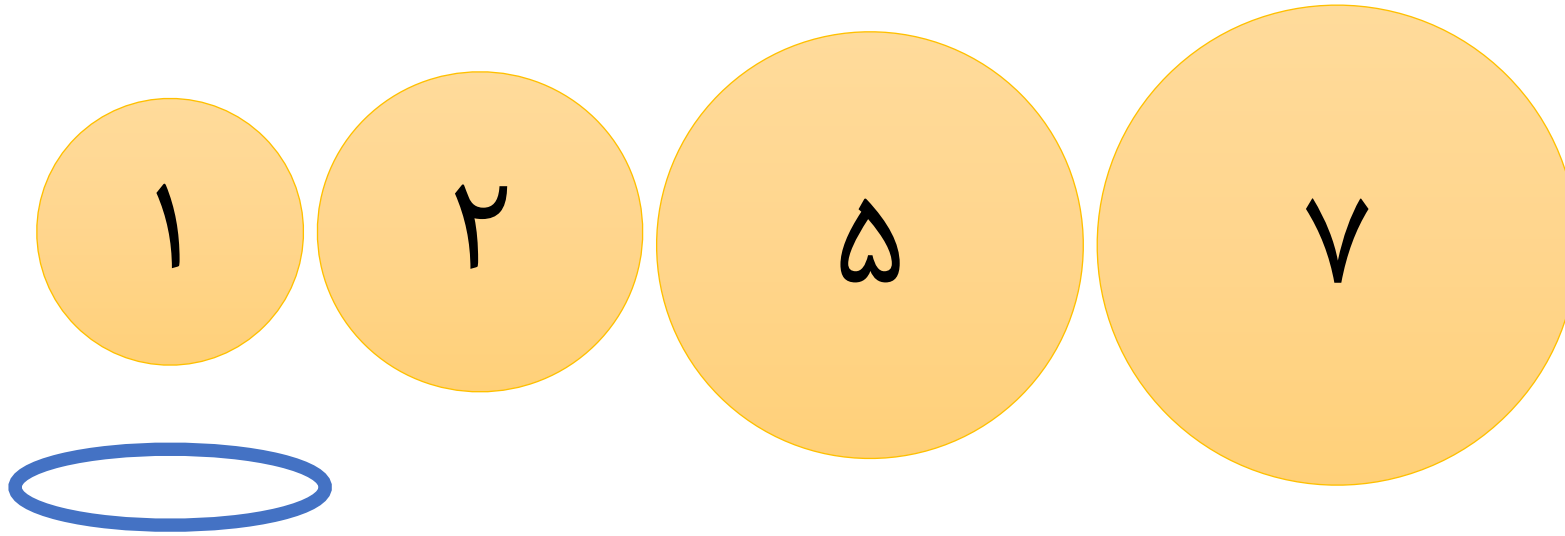
					۵	۴	۳	۲	۱	♦	پول
					۱	۲	۲	۱	۱	♦	تعداد سکه

بیاید مساله را از ابتدا حل کنیم.



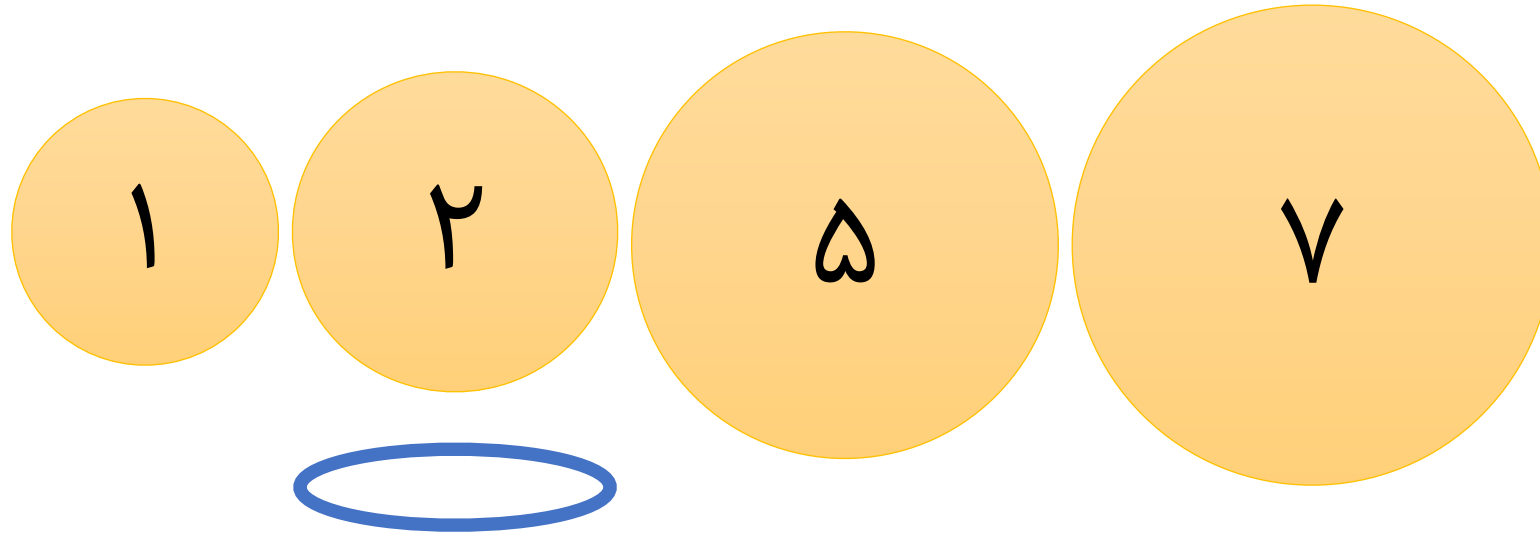
				۶	۵	۴	۳	۲	۱	♦	پول
					۱	۲	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



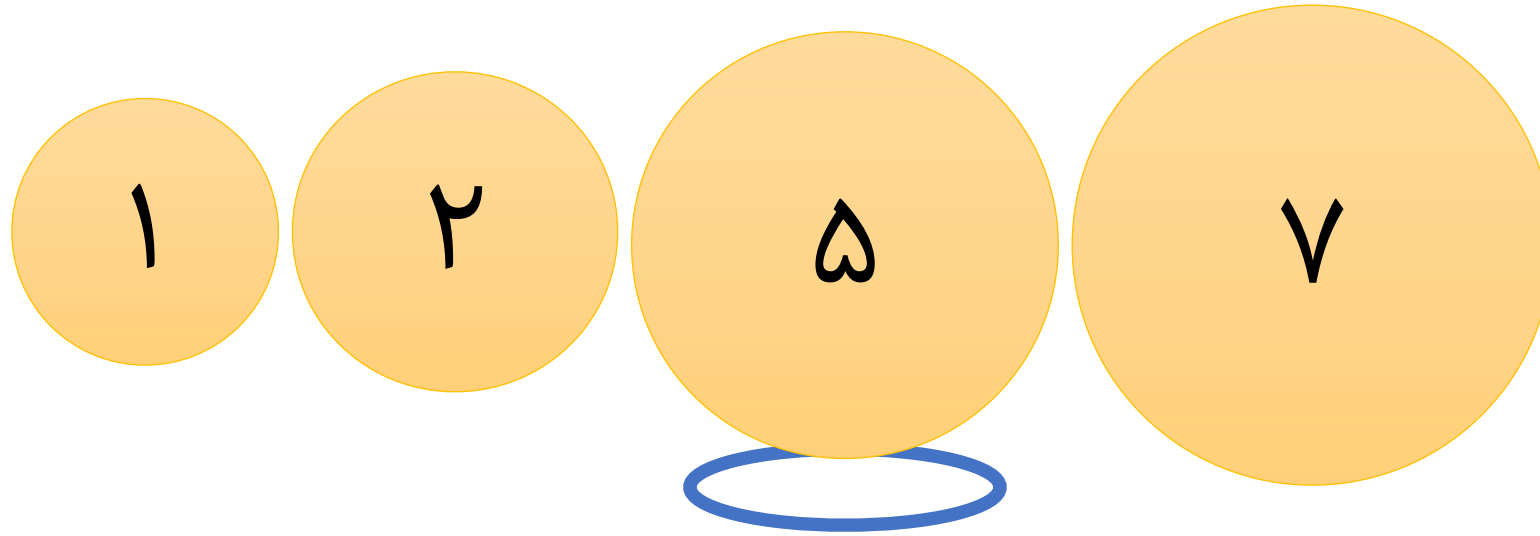
				۶	۵	۴	۳	۲	۱	♦	پول
				۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



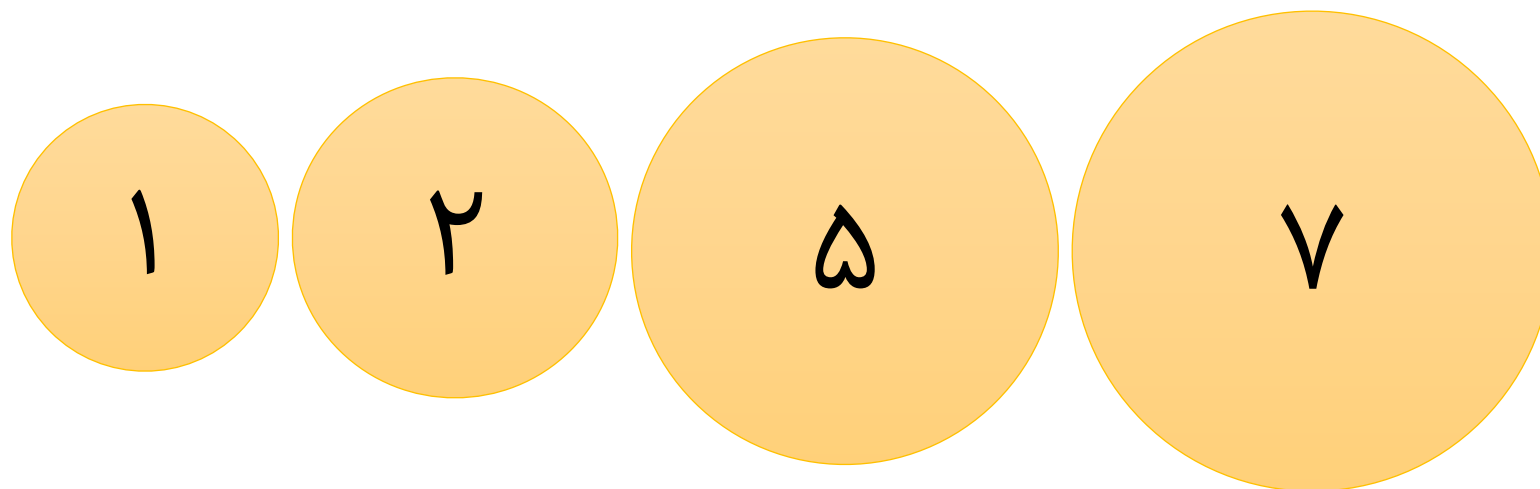
				۶	۵	۴	۳	۲	۱	♦	پول
				۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیابید مساله را از ابتدا حل کنیم.



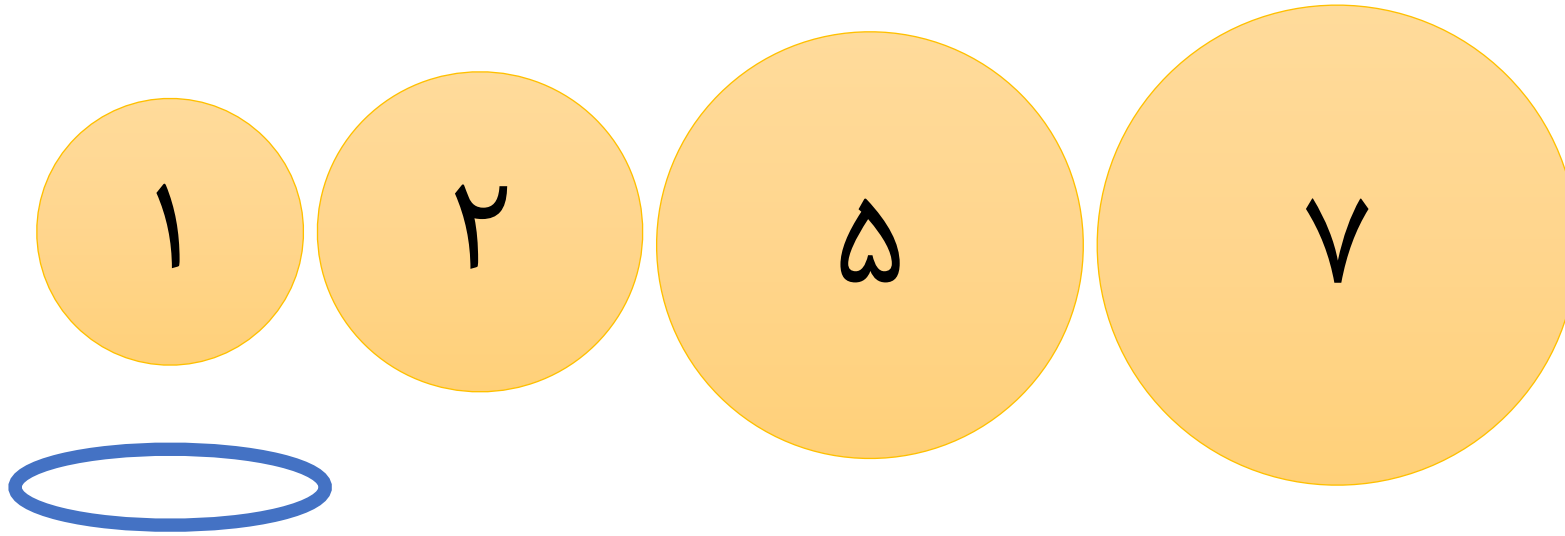
				۶	۵	۴	۳	۲	۱	♦	پول
				۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیاپید مساله را از ابتدا حل کنیم.



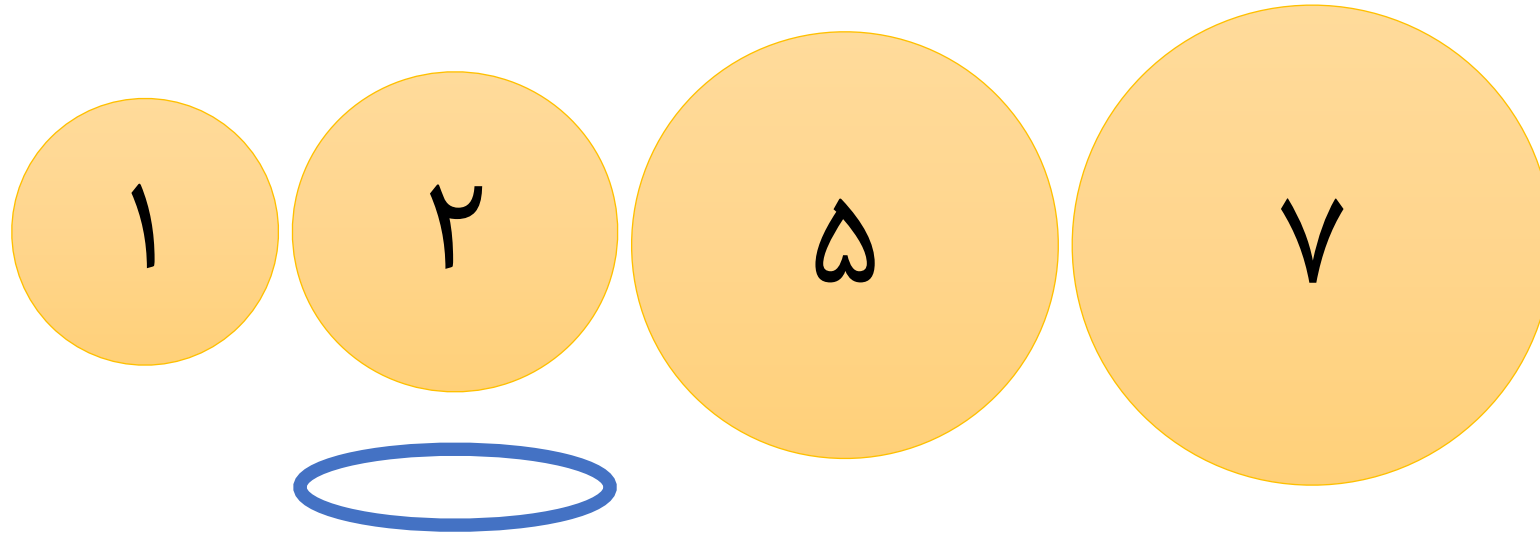
			۷	۶	۵	۴	۳	۲	۱	♦	پول
				۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



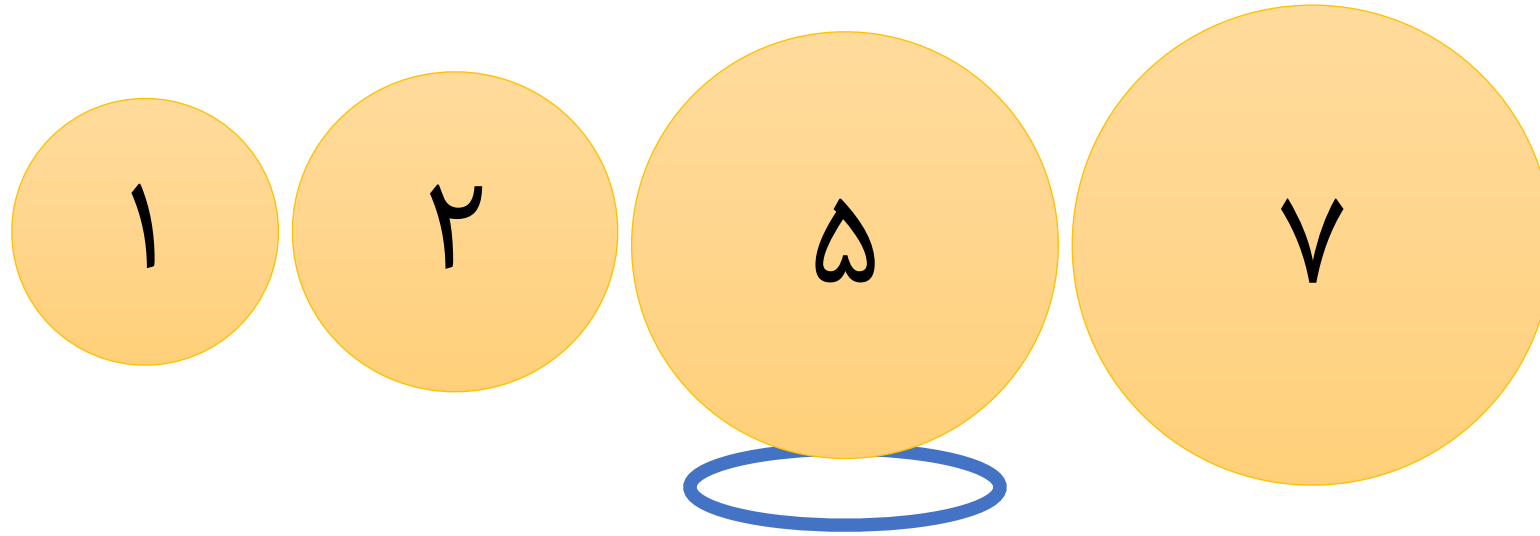
			۷	۶	۵	۴	۳	۲	۱	♦	پول
			۳	۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



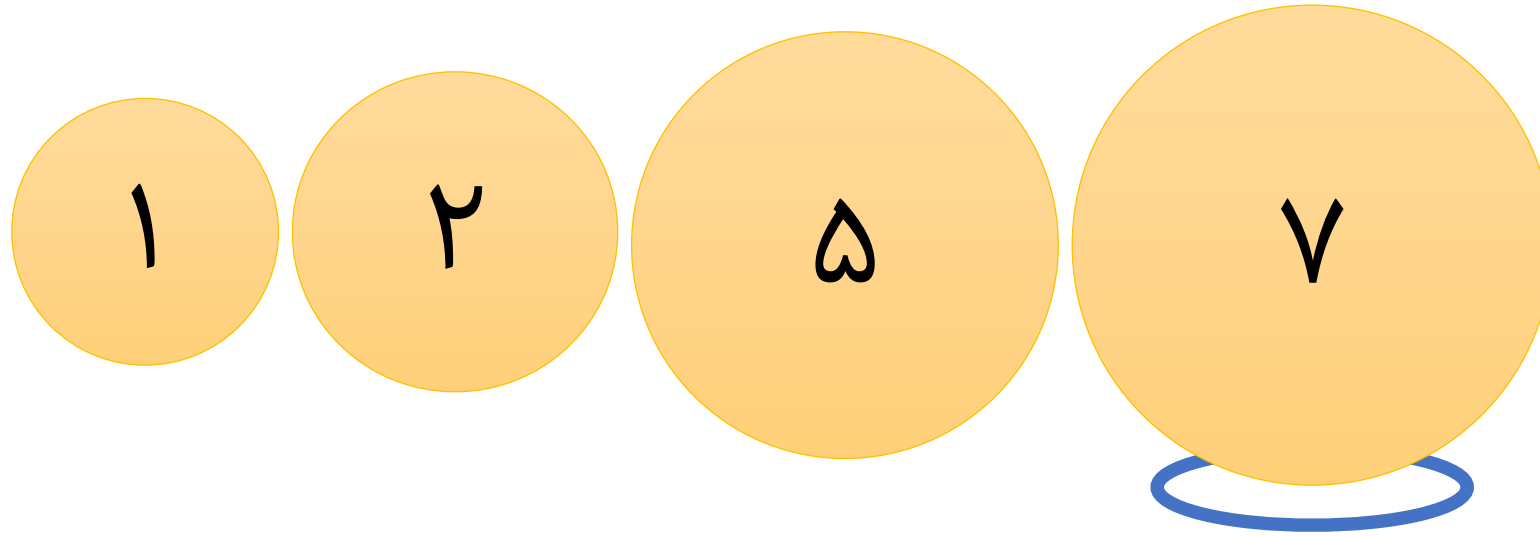
			۷	۶	۵	۴	۳	۲	۱	♦	پول
			۲	۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



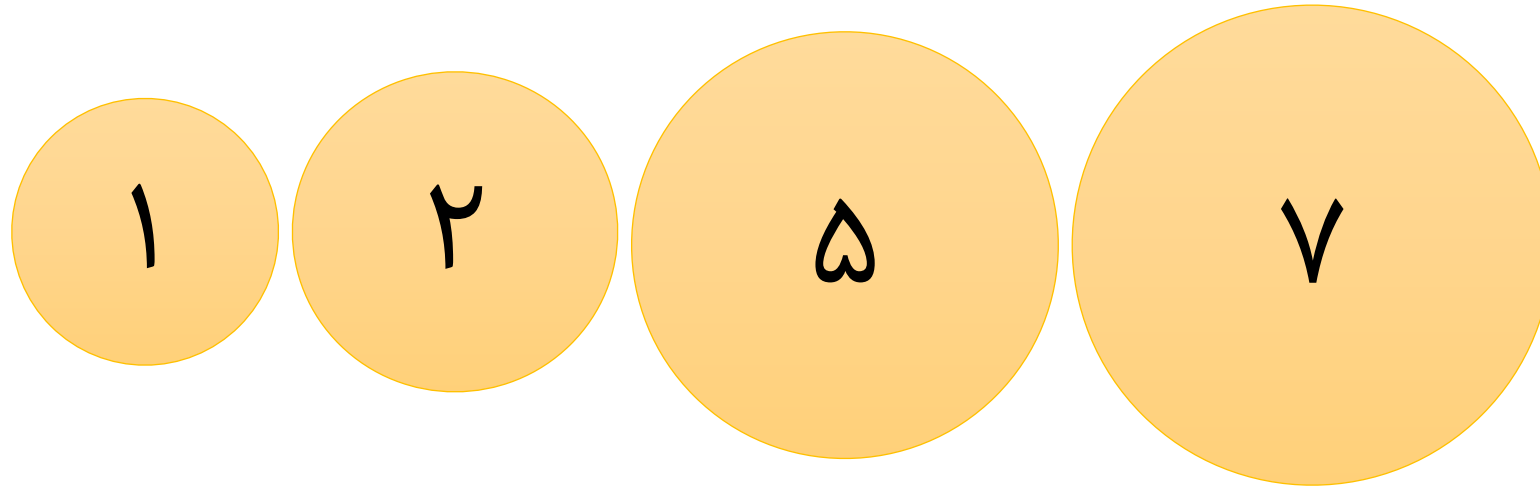
			۷	۶	۵	۴	۳	۲	۱	♦	پول
			۲	۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



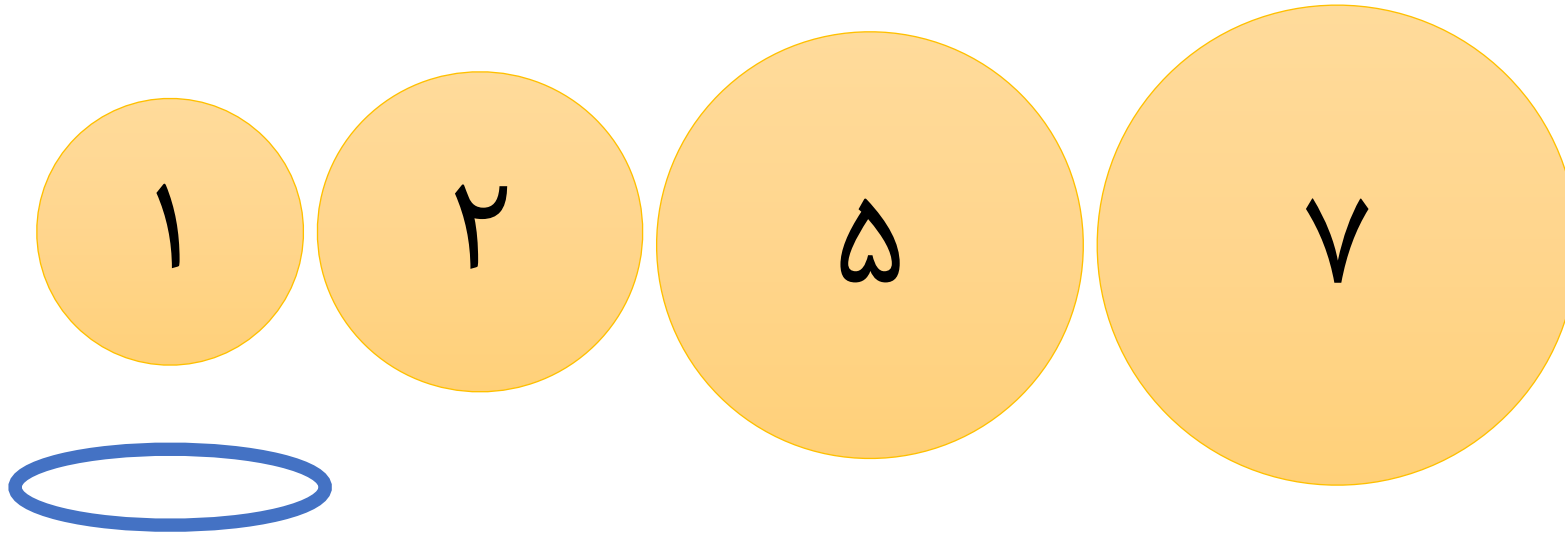
			۷	۶	۵	۴	۳	۲	۱	♦	پول
			۱	۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



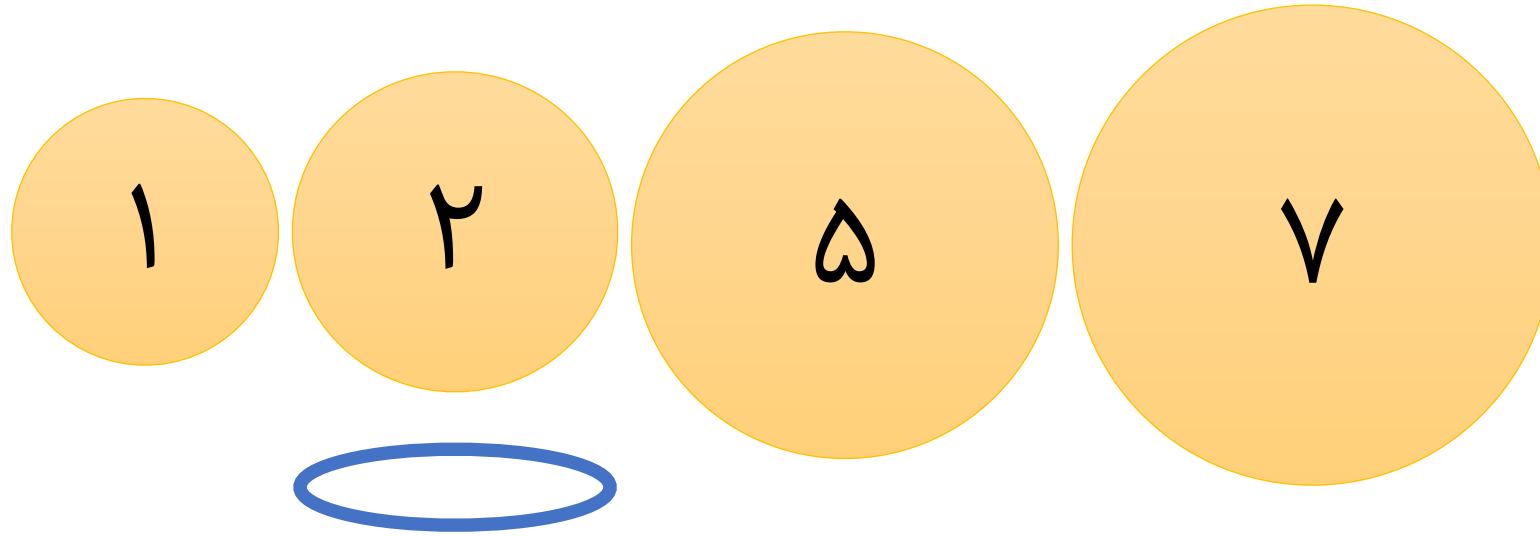
		۸	۷	۶	۵	۴	۳	۲	۱	♦	پول
			۱	۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



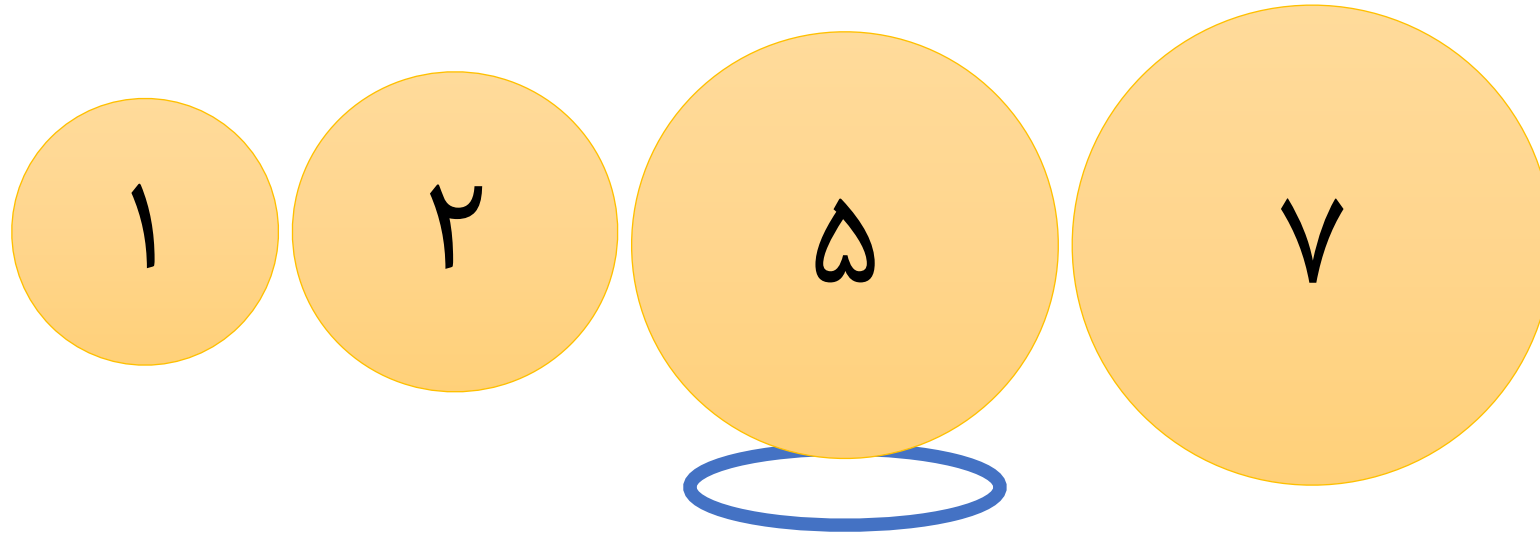
		۸	۷	۶	۵	۴	۳	۲	۱	♦	پول
		۲	۱	۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



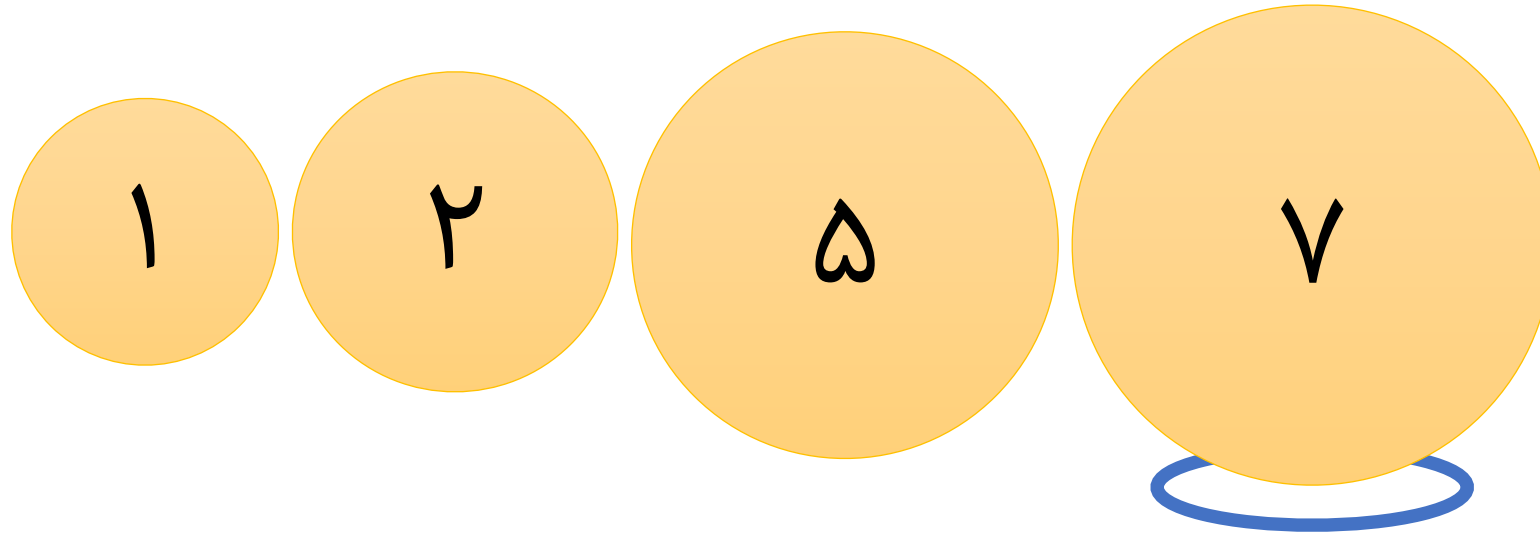
		۸	۷	۶	۵	۴	۳	۲	۱	♦	پول
		۲	۱	۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



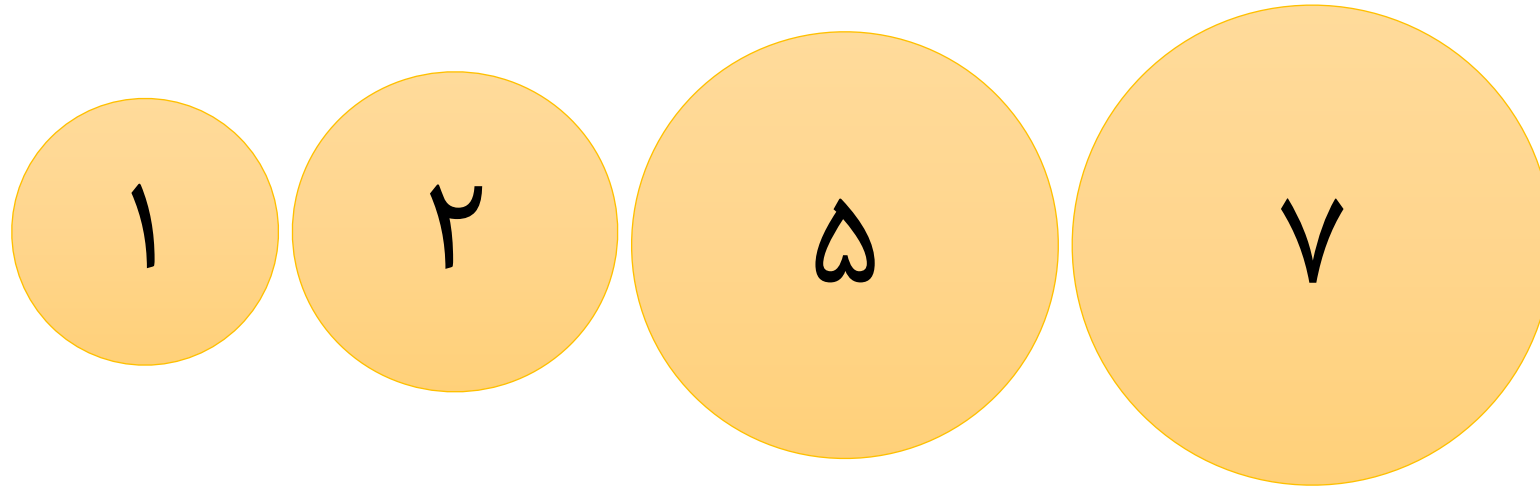
		۸	۷	۶	۵	۴	۳	۲	۱	♦	پول
		۲	۱	۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



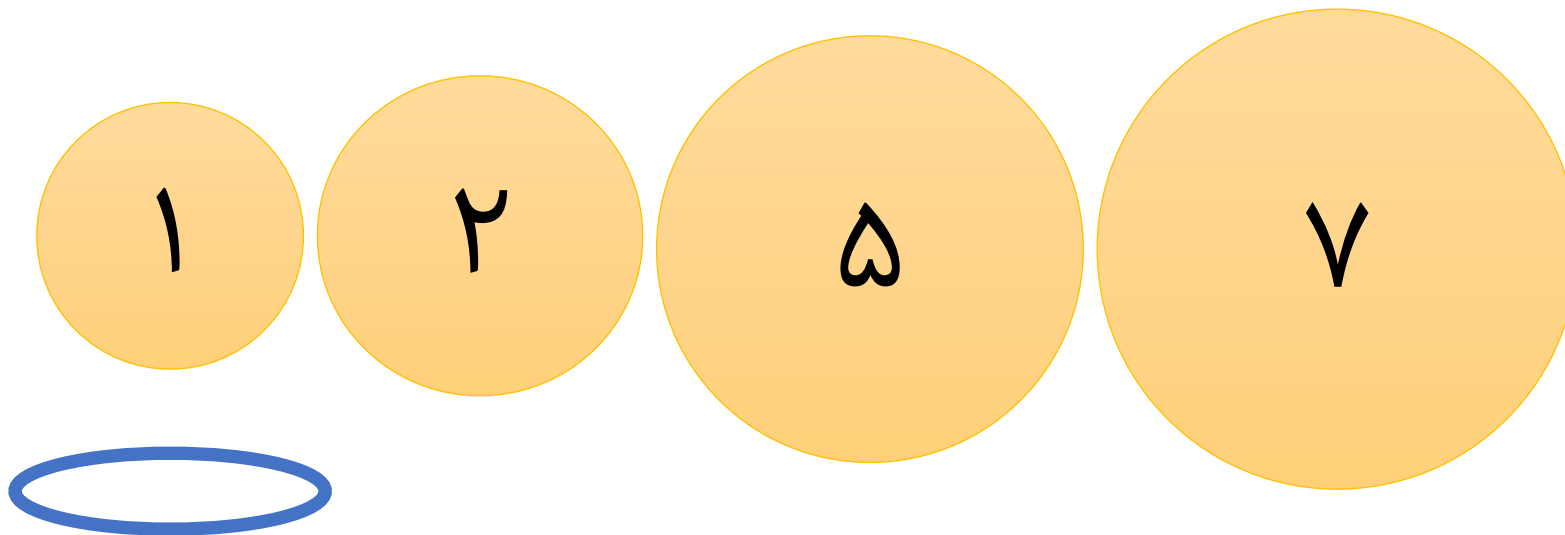
		۸	۷	۶	۵	۴	۳	۲	۱	♦	پول
		۲	۱	۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



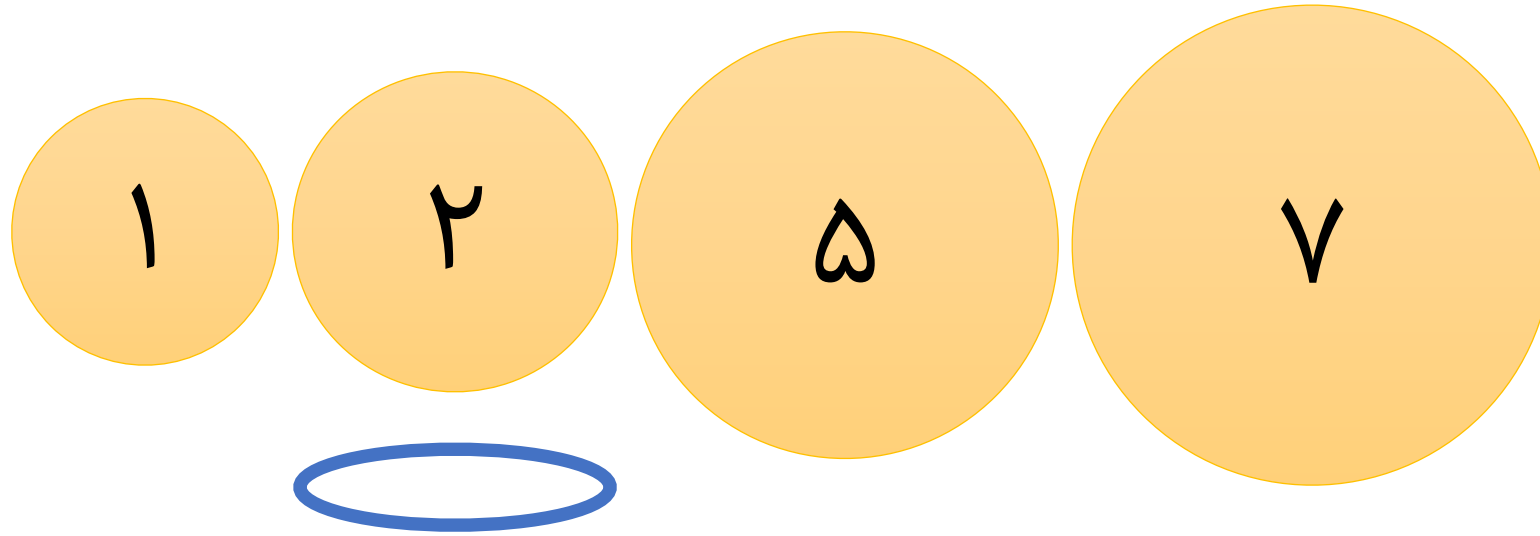
	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰	پول
تعداد سکه		۲	۱	۲	۱	۲	۲	۱	۱	۰	

بیا یاد مساله را از ابتدا حل کنیم.



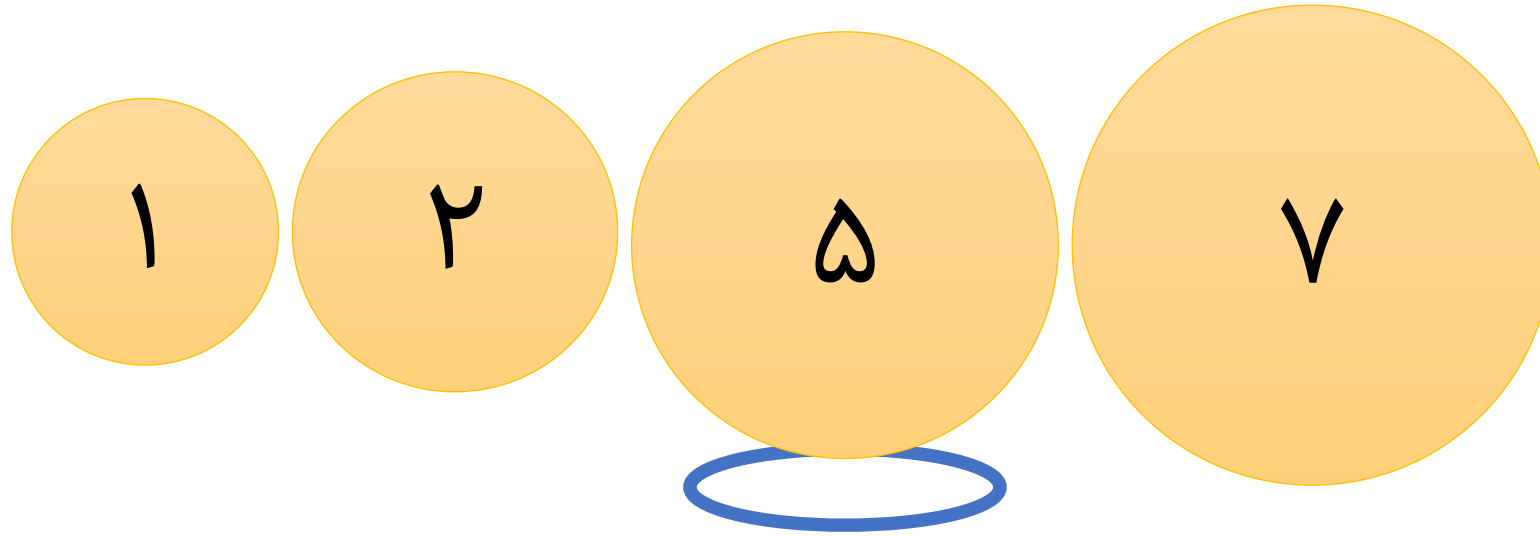
	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰	پول
	۳	۲	۱	۲	۱	۲	۲	۱	۱	۰	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



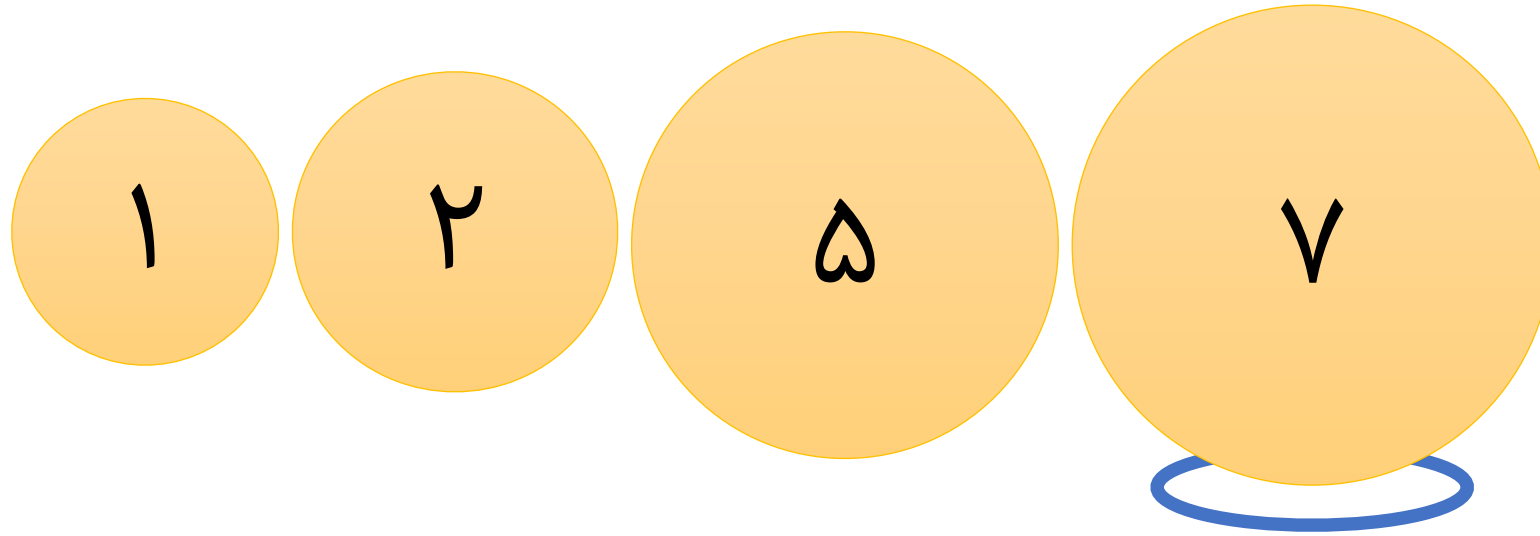
	۹	۸	۷	۶	۵	۴	۳	۲	۱	♦	پول
	۲	۲	۱	۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیابید مساله را از ابتدا حل کنیم.



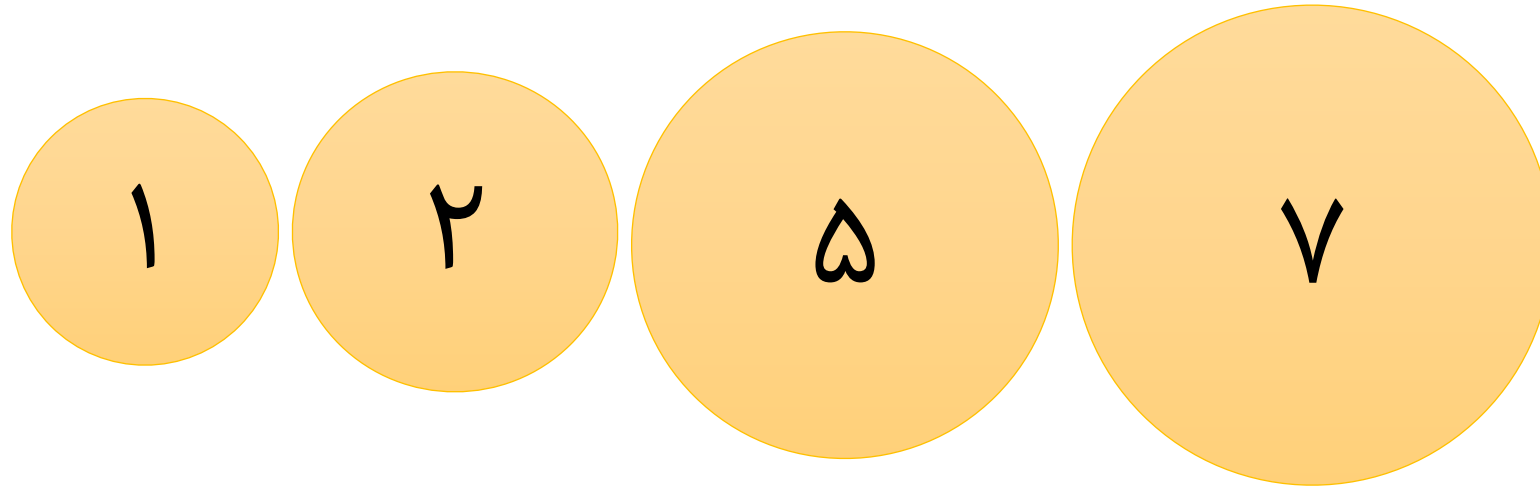
	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰	پول
	۲	۲	۱	۲	۱	۲	۲	۱	۱	۰	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



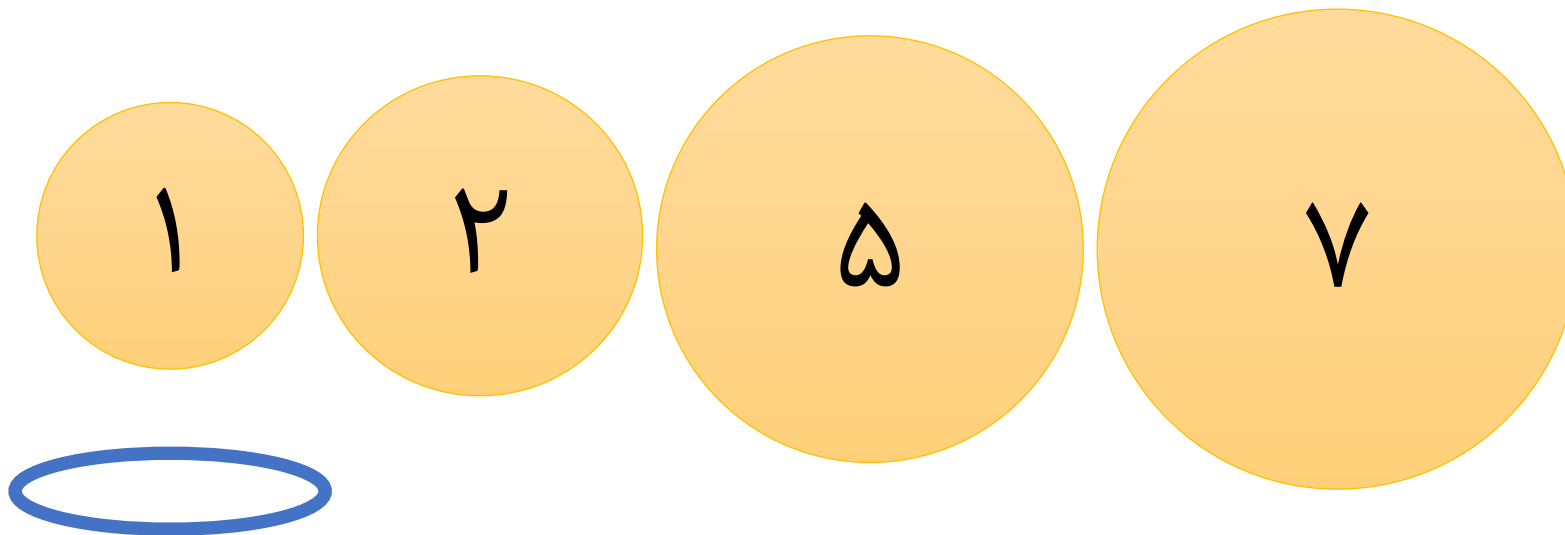
	۹	۸	۷	۶	۵	۴	۳	۲	۱	♦	پول
	۲	۲	۱	۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیاید مساله را از ابتدا حل کنیم.



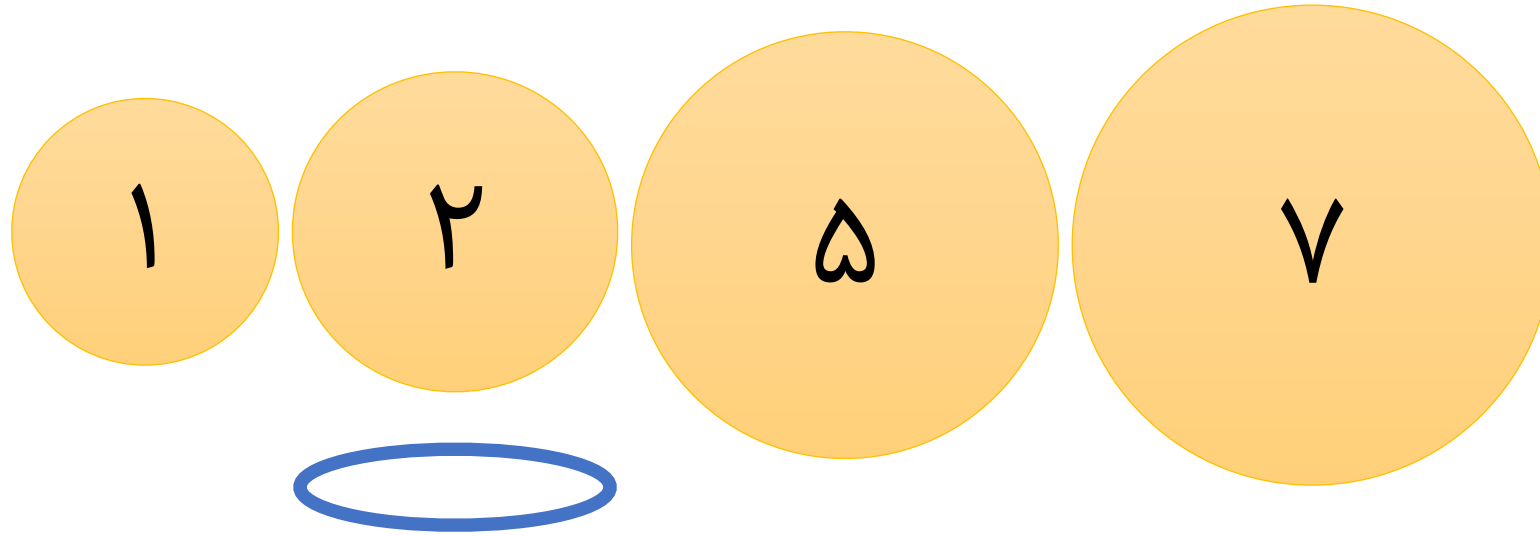
۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰	پول
	۲	۲	۱	۲	۱	۲	۲	۱	۱	۰	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



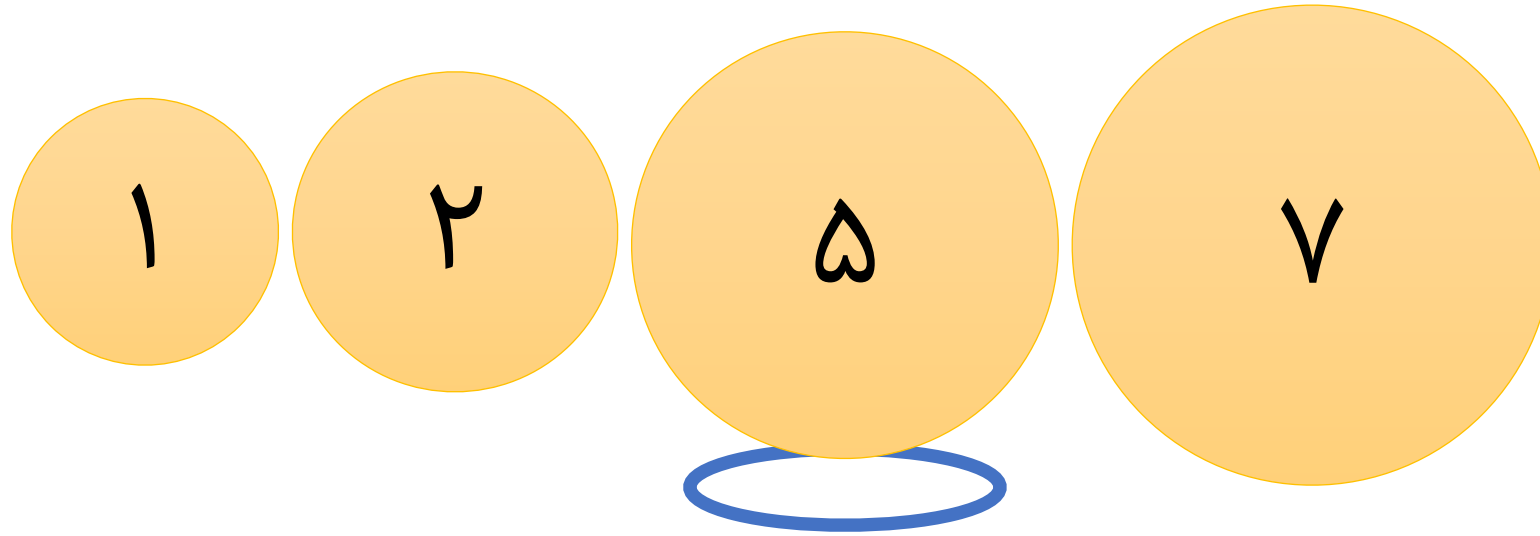
۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰	پول
۳	۲	۲	۱	۲	۱	۲	۲	۱	۱	۰	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



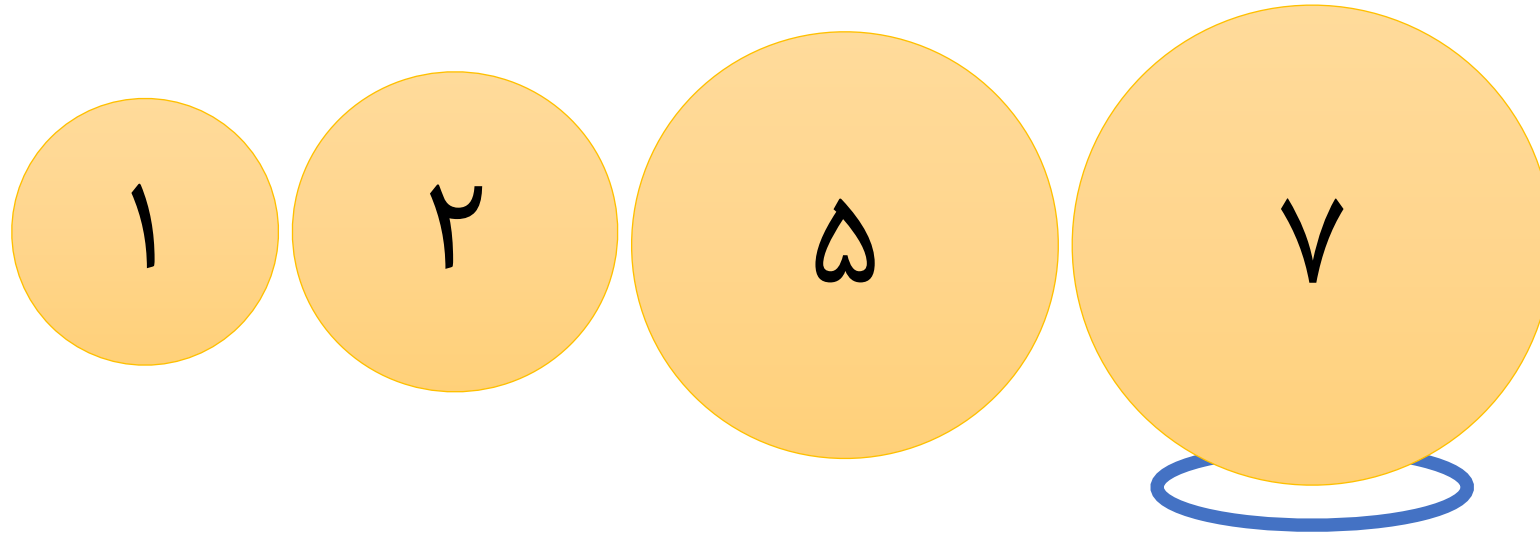
۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰	پول
۳	۲	۲	۱	۲	۱	۲	۲	۱	۱	۰	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



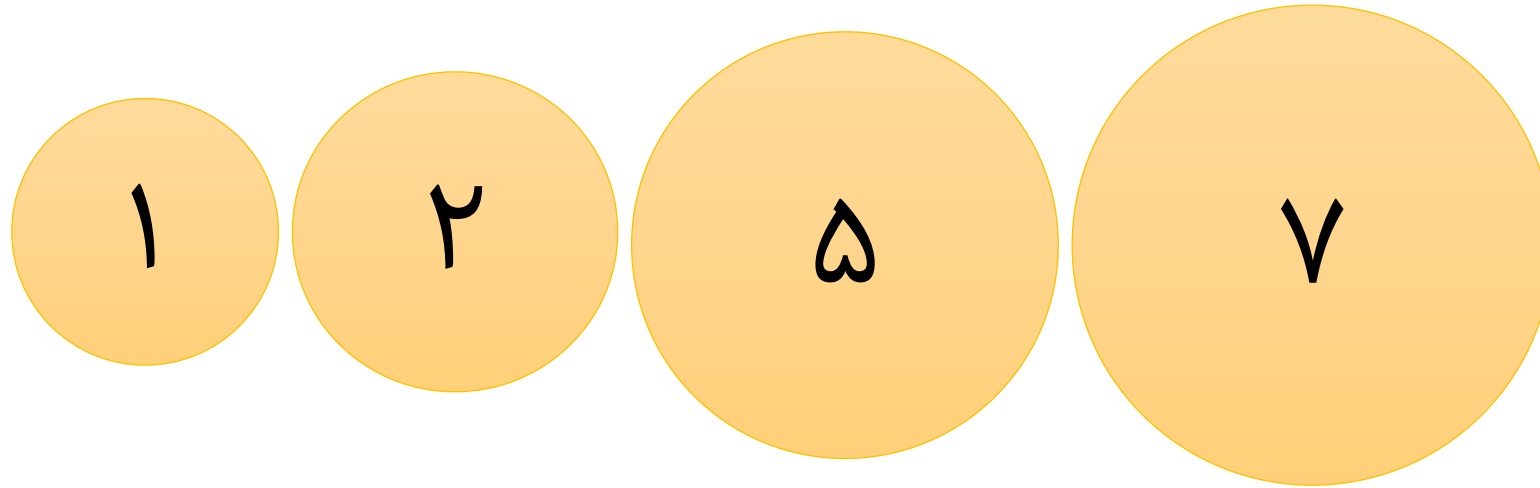
۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰	پول
۲	۲	۲	۱	۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیا یاد مساله را از ابتدا حل کنیم.



۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	♦	پول
۲	۲	۲	۱	۲	۱	۲	۲	۱	۱	♦	تعداد سکه

بیاید مساله را از ابتدا حل کنیم.



۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰	پول
۲	۲	۲	۱	۲	۱	۲	۲	۱	۱	۰	تعداد سکه

الگوریتم یافتن کمترین تعداد سکه‌ها

• تعداد سکه‌ها n ، پول موردنظر m و مقدار سکه‌ها c را بخوان.



الگوریتم یافتن کمترین تعداد سکه‌ها

- تعداد سکه‌ها n ، پول موردنظر m و مقدار سکه‌ها c را بخوان.
- آرایه‌ی s را به طول $m+1$ با مقادیر بی‌نهایت در نظر بگیر.



الگوریتم یافتن کمترین تعداد سکه‌ها

- تعداد سکه‌ها n ، پول موردنظر m و مقدار سکه‌ها c را بخوان.
- آرایه‌ی s را به طول $m+1$ با مقادیر بی‌نهایت در نظر بگیر.
- $s[0]=0$

الگوریتم یافتن کمترین تعداد سکه‌ها

- تعداد سکه‌ها n ، پول موردنظر m و مقدار سکه‌ها c را بخوان.
- آرایه‌ی s را به طول $m+1$ با مقادیر بی‌نهایت در نظر بگیر.
- $s[0]=0$
- به ازای هر i بین 1 تا m

الگوریتم یافتن کمترین تعداد سکه‌ها

- تعداد سکه‌ها n ، پول موردنظر m و مقدار سکه‌ها c را بخوان.
- آرایه‌ی s را به طول $m+1$ با مقادیر بی‌نهایت در نظر بگیر.
- $s[0]=0$
- به ازای هر i بین 1 تا m
- به ازای هر j بین 0 تا $n-1$

الگوریتم یافتن کمترین تعداد سکه‌ها

- تعداد سکه‌ها n ، پول موردنظر m و مقدار سکه‌ها c را بخوان.
- آرایه‌ی s را به طول $m+1$ با مقادیر بی‌نهایت در نظر بگیر.
- $s[0]=0$
- به ازای هر i بین 1 تا m
به ازای هر j بین 0 تا $n-1$
اگر $c[j] \leq i$

الگوریتم یافتن کمترین تعداد سکه‌ها

- تعداد سکه‌ها n ، پول موردنظر m و مقدار سکه‌ها c را بخوان.
- آرایه‌ی s را به طول $m+1$ با مقادیر بی‌نهایت در نظر بگیر.
- $s[0]=0$
- به ازای هر i بین 1 تا m

به ازای هر j بین 0 تا $n-1$

اگر $c[j] \leq i$

$$k = i - c[j]$$



الگوریتم یافتن کمترین تعداد سکه‌ها

- تعداد سکه‌ها n ، پول موردنظر m و مقدار سکه‌ها c را بخوان.
- آرایه‌ی s را به طول $m+1$ با مقادیر بی‌نهایت در نظر بگیر.
- $s[0]=0$
- به ازای هر i بین 1 تا m

به ازای هر j بین 0 تا $n-1$

اگر $c[j] \leq i$

$$k = i - c[j]$$

$$\text{اگر } s[k] + 1 < s[i]$$

الگوریتم یافتن کمترین تعداد سکه‌ها

- تعداد سکه‌ها n ، پول موردنظر m و مقدار سکه‌ها c را بخوان.
- آرایه‌ی s را به طول $m+1$ با مقادیر بی‌نهایت در نظر بگیر.
- $s[0]=0$
- به ازای هر i بین 1 تا m

به ازای هر j بین 0 تا $n-1$

اگر $c[j] \leq i$

$$k = i - c[j]$$

اگر $s[k] + 1 < s[i]$

$$s[i] = s[k] + 1$$



الگوریتم یافتن کمترین تعداد سکه‌ها

- تعداد سکه‌ها n ، پول موردنظر m و مقدار سکه‌ها c را بخوان.
- آرایه‌ی s را به طول $m+1$ با مقادیر بی‌نهایت در نظر بگیر.

$$s[0] = 0$$

- به ازای هر i بین 1 تا m

به ازای هر j بین 0 تا $n-1$

اگر $c[j] \leq i$

$$k = i - c[j]$$

اگر $s[k] + 1 < s[i]$

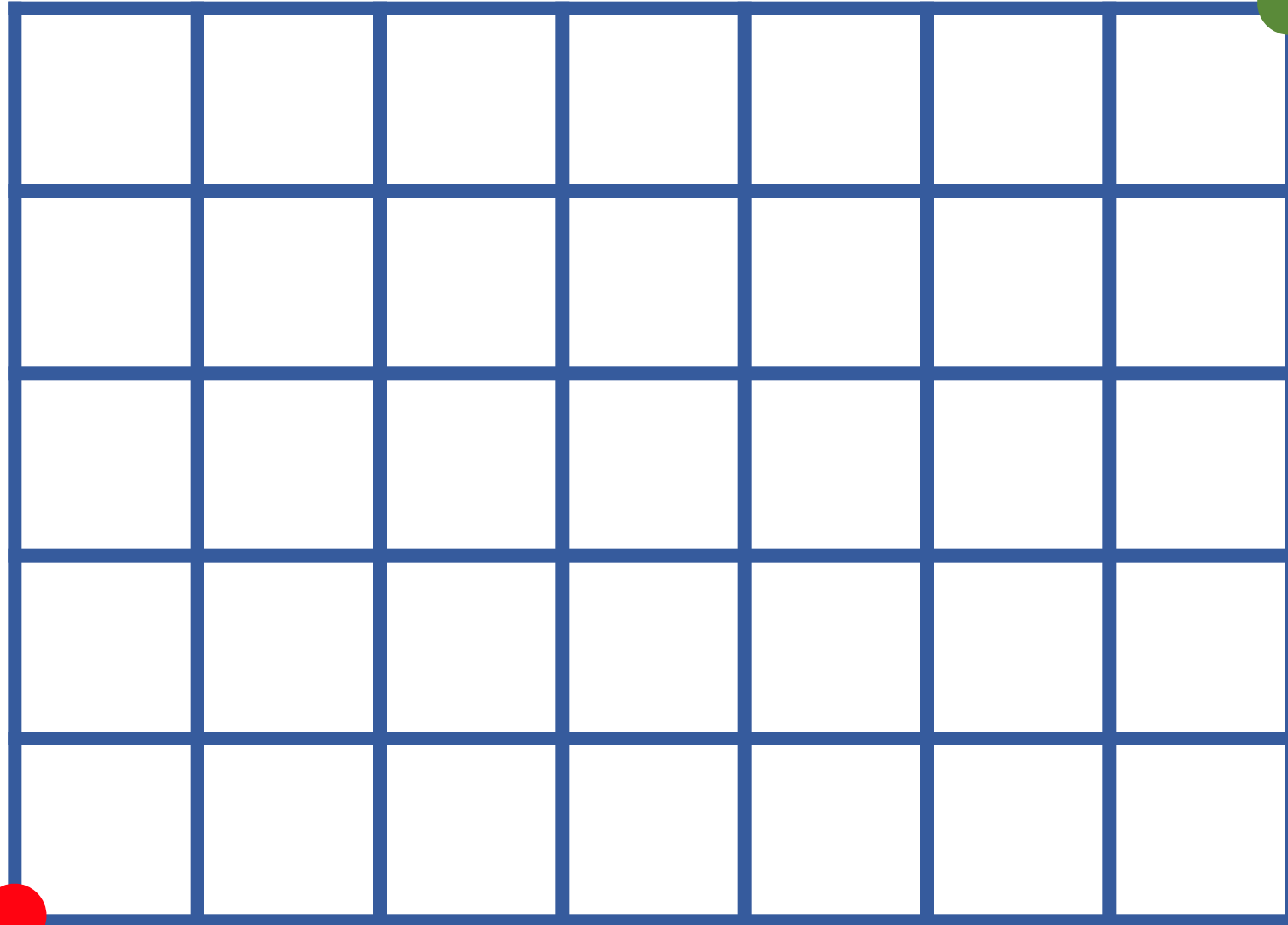
$$s[i] = s[k] + 1$$

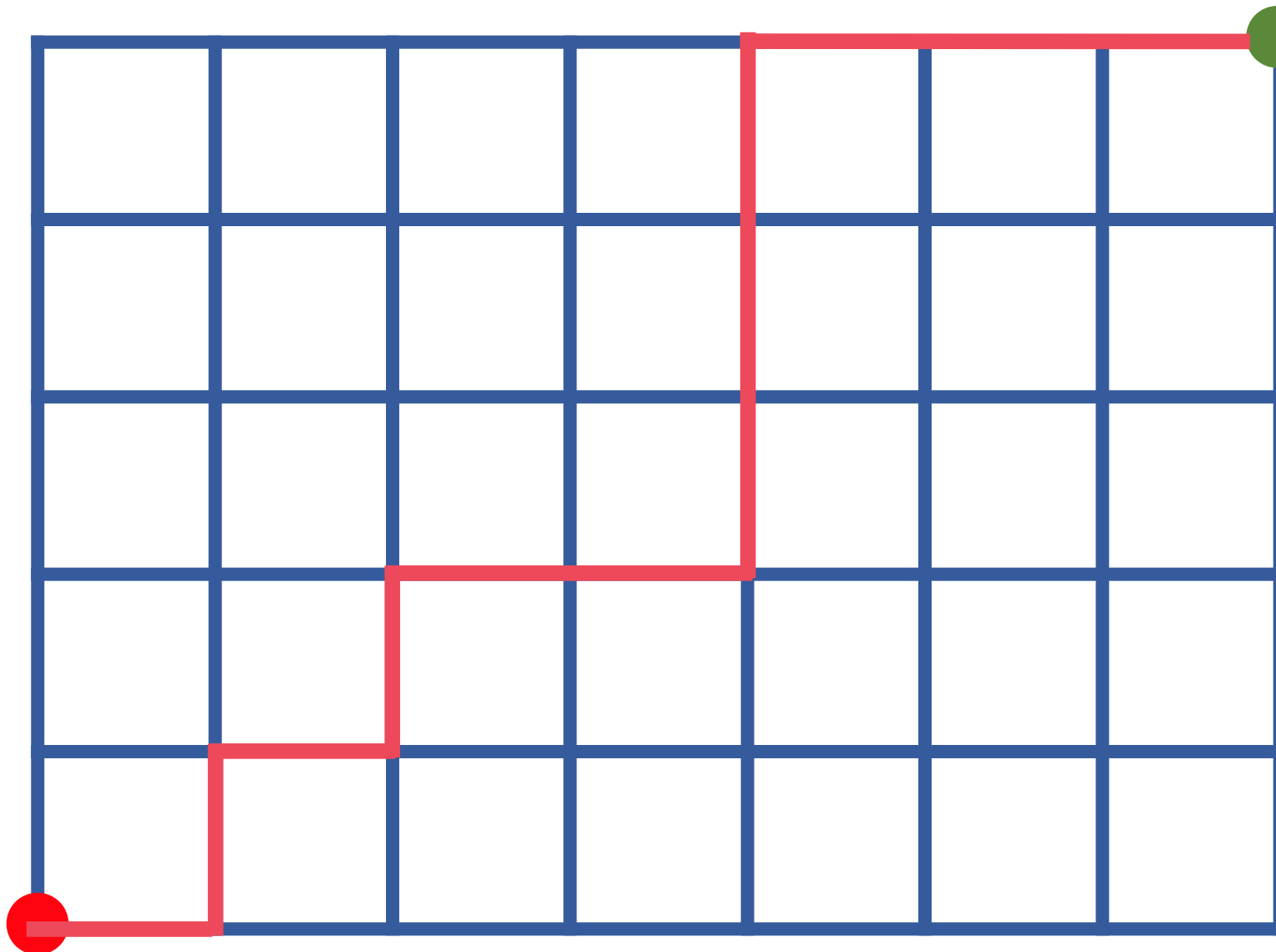
- s را چاپ کن.



وقت كدزدن (:

ببراس تنوع طلب





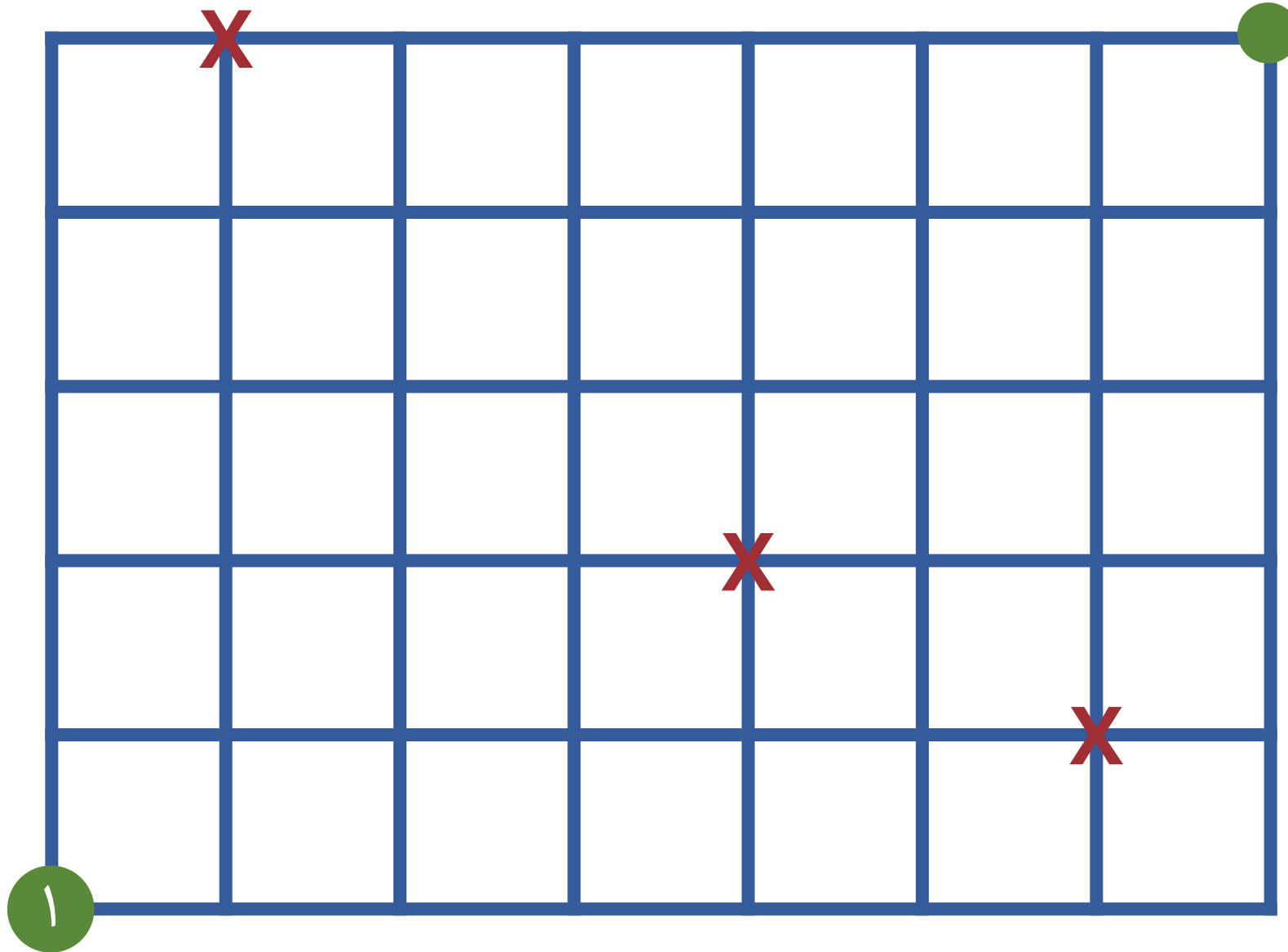
تعداد زیرمجموعه‌های r عضوی از مجموعه‌ی n عضوی

$$C(n, r) = C(n, n - r) = \frac{n!}{(n - r)!r!}$$

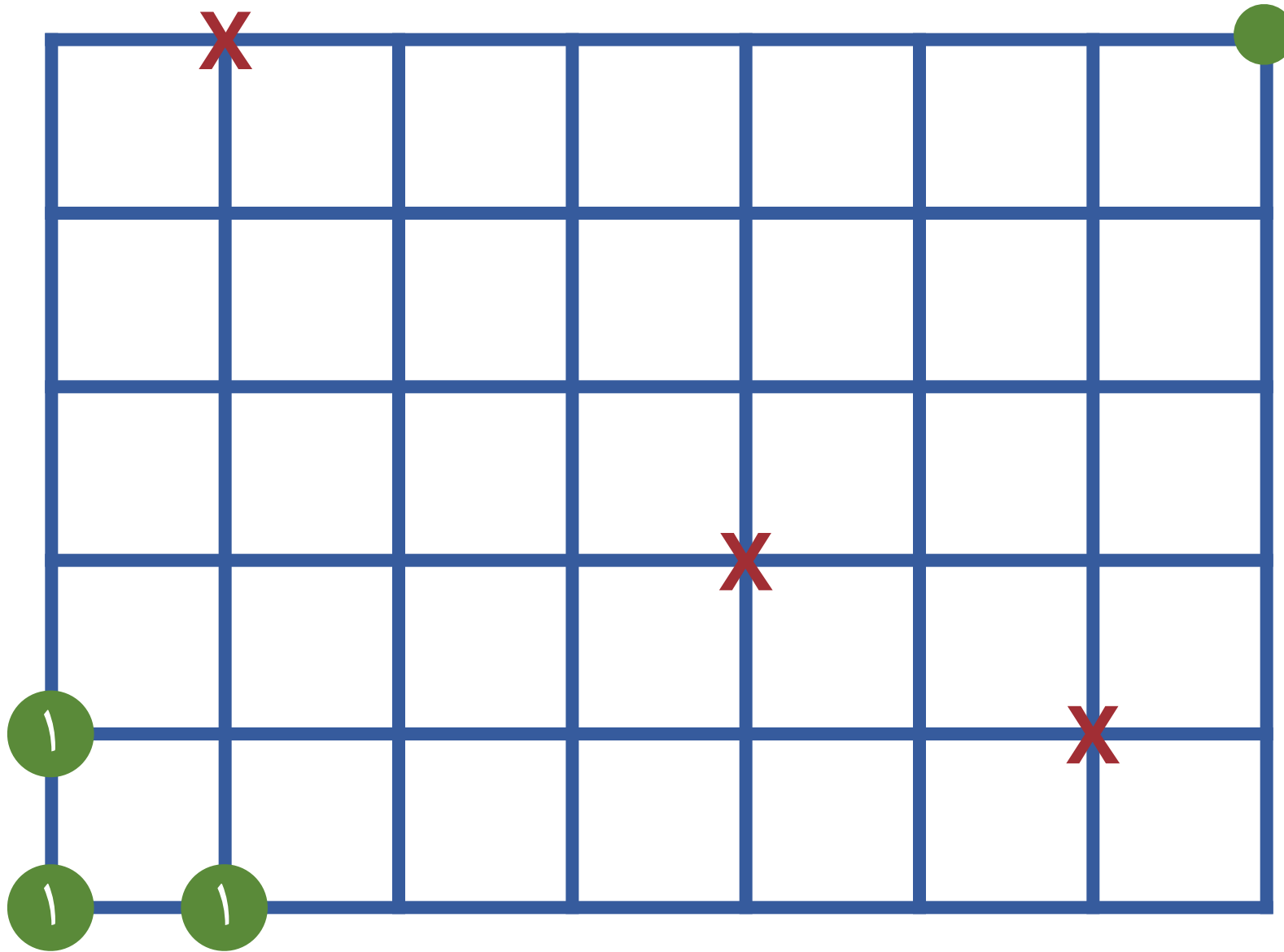
اگر بعضی تقاطعها مسدود باشد چطور تعداد مسیرها را بیابیم؟

A 5x6 grid with a red dot at the bottom-left corner, a green dot at the top-right corner, and three red 'X' marks at (1,2), (4,4), and (5,5) representing obstacles. A cartoon squirrel is at the red dot and a house icon is at the green dot.

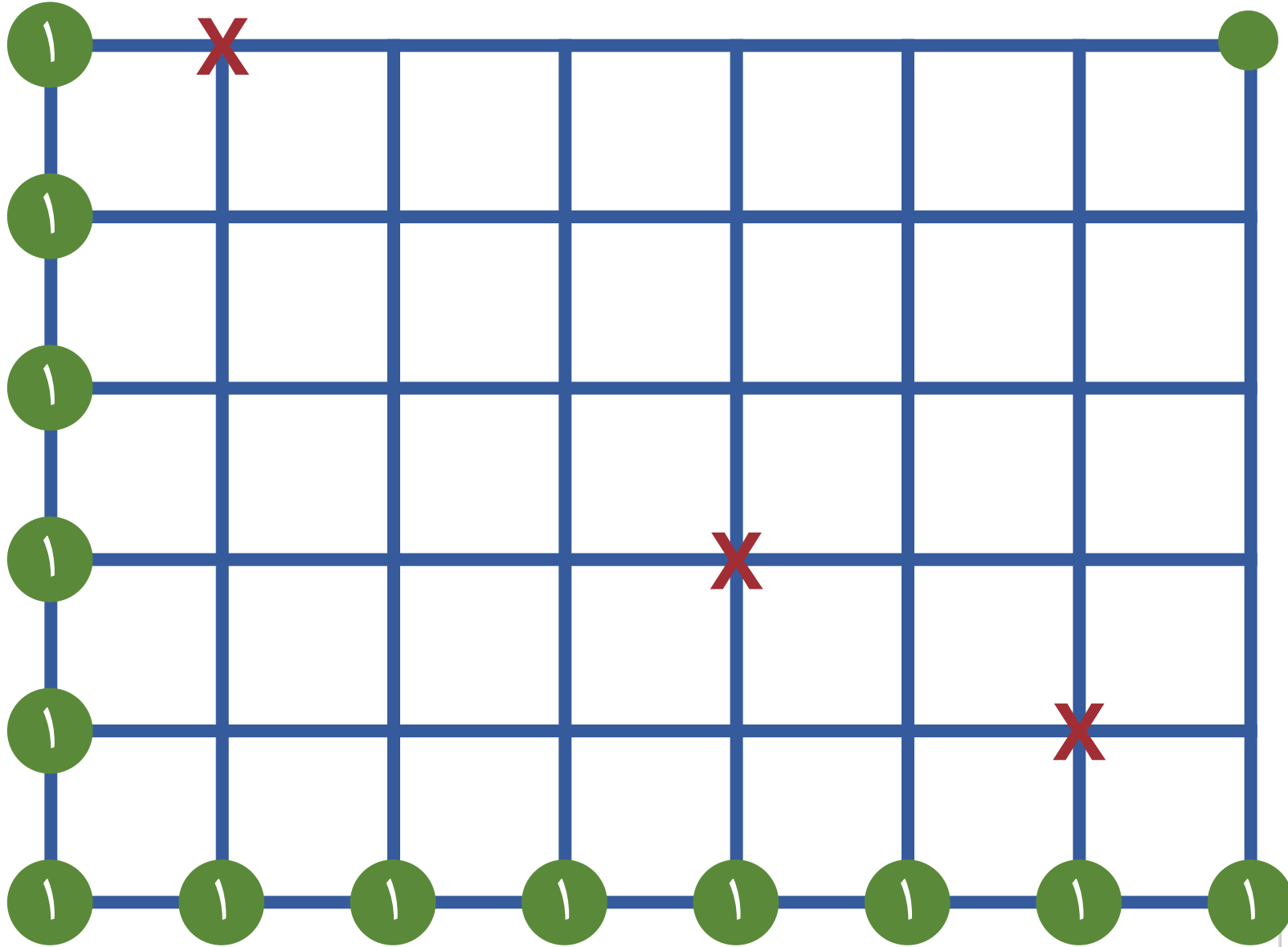
اگر بعضی تقاطعها مسدود باشد چطور تعداد مسیرها را بیابیم؟



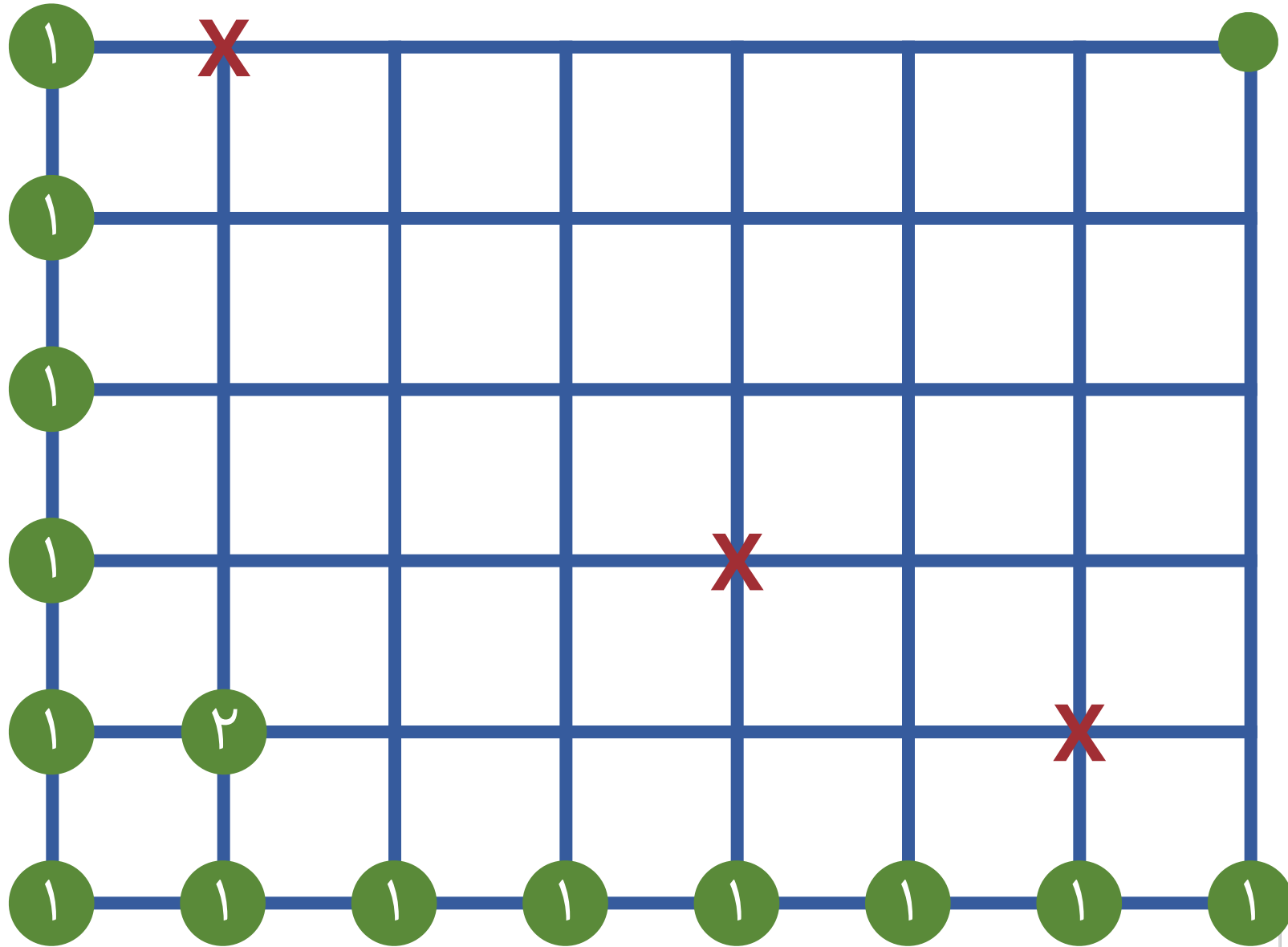
اگر بعضی تقاطعها مسدود باشد چطور تعداد مسیرها را بیابیم؟



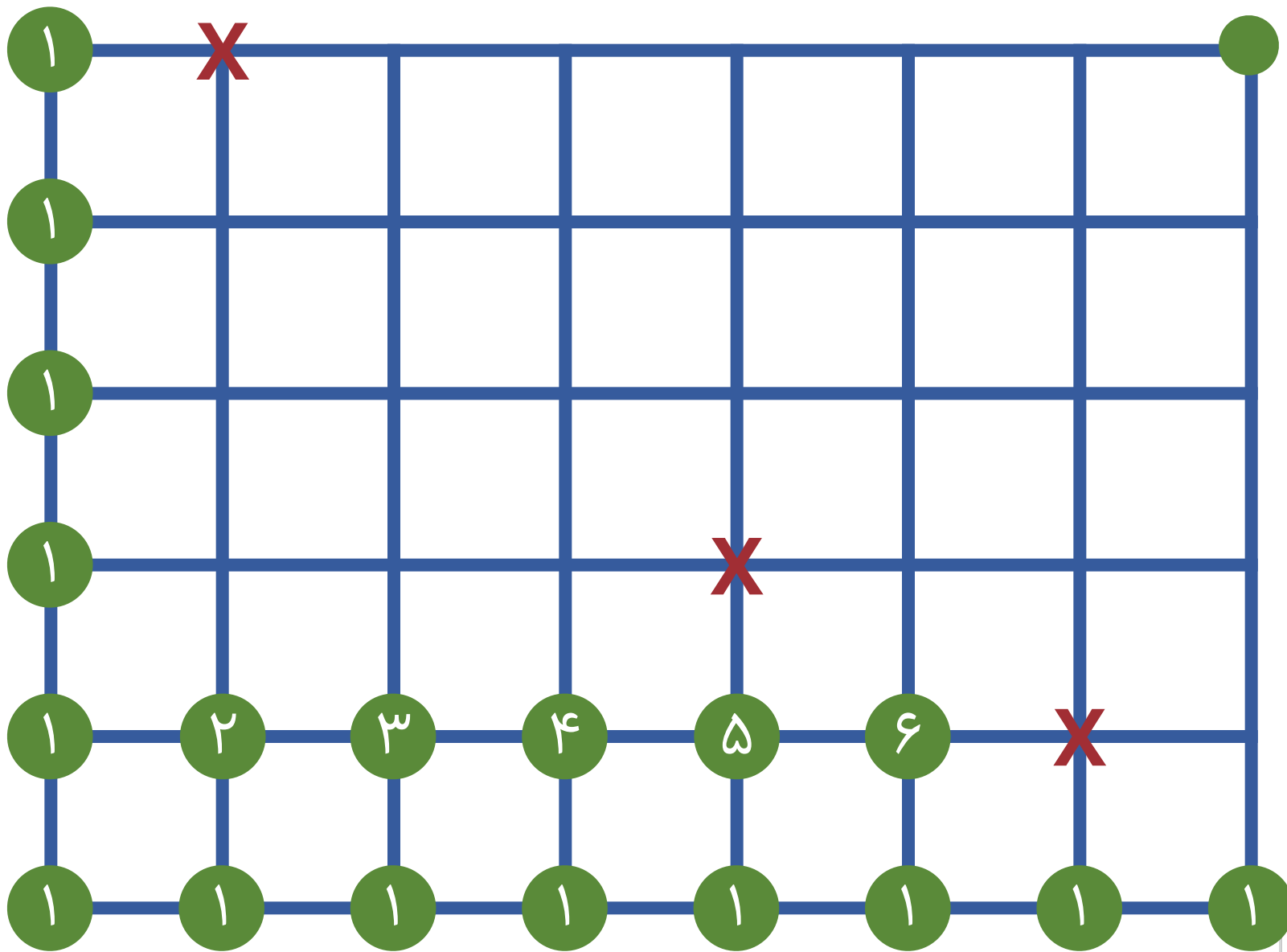
اگر بعضی تقاطعها مسدود باشد چطور تعداد مسیرها را بیابیم؟



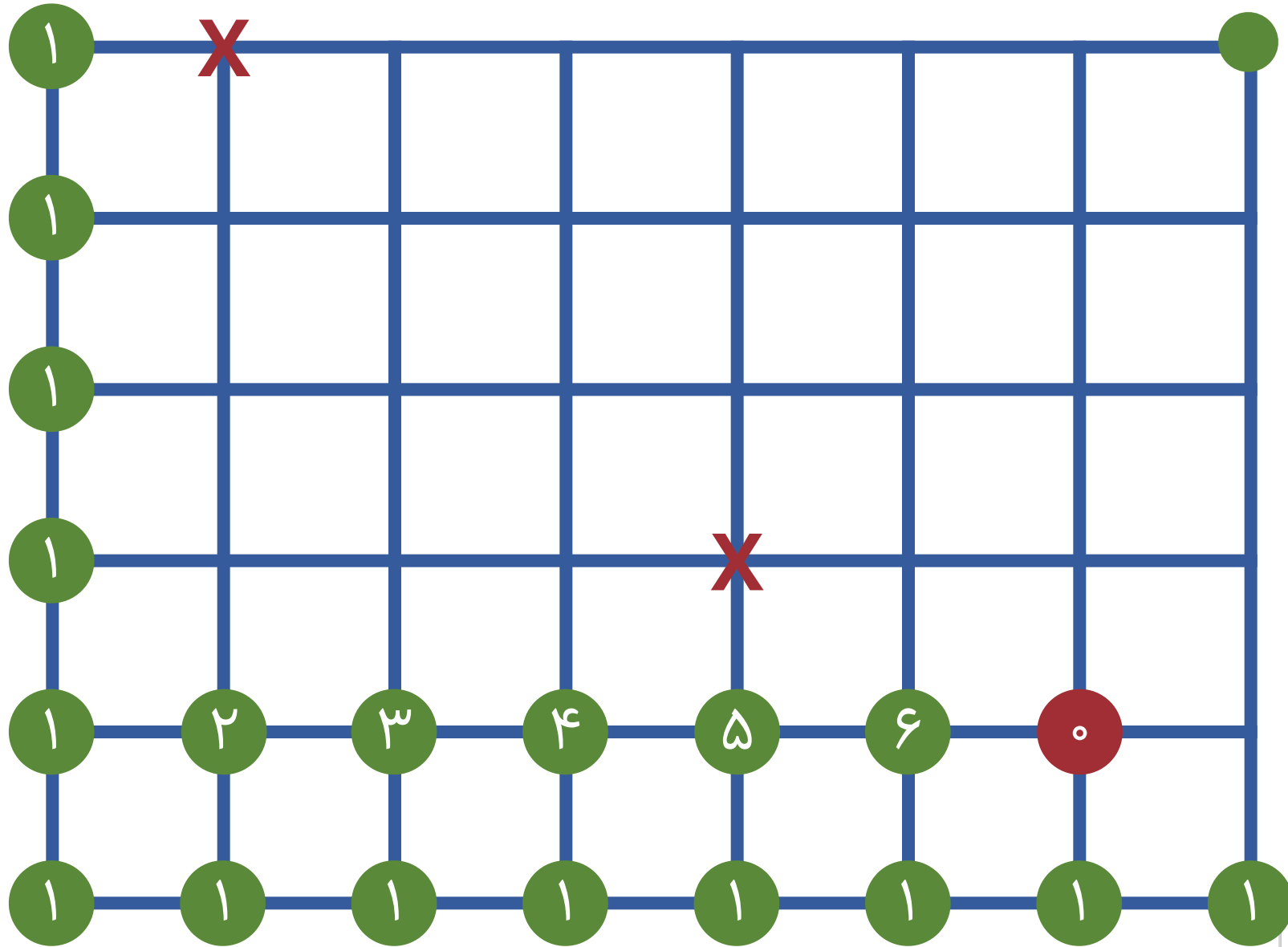
اگر بعضی تقاطعها مسدود باشد چطور تعداد مسیرها را بیابیم؟



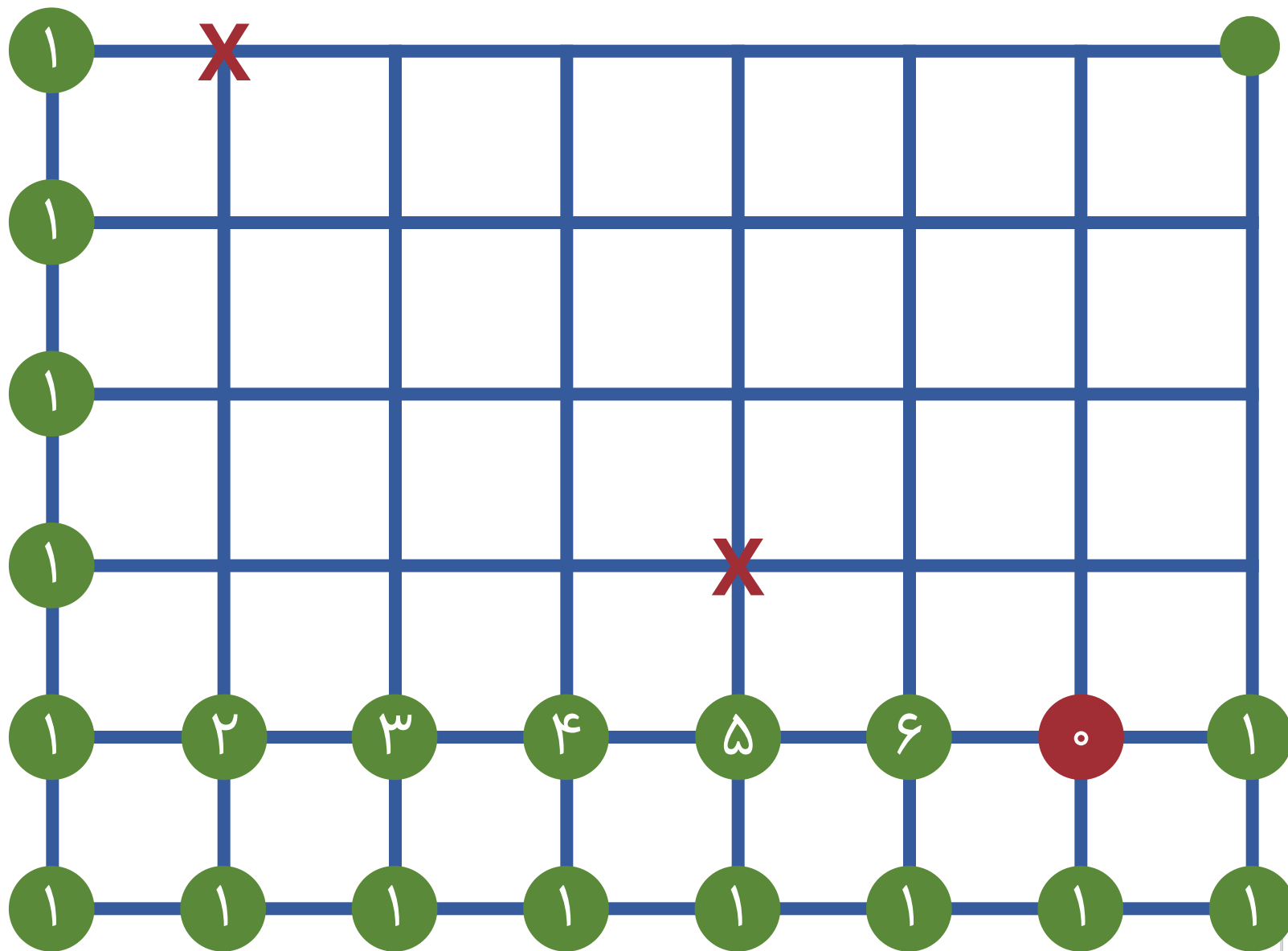
اگر بعضی تقاطعها مسدود باشد چطور تعداد مسیرها را بیابیم؟



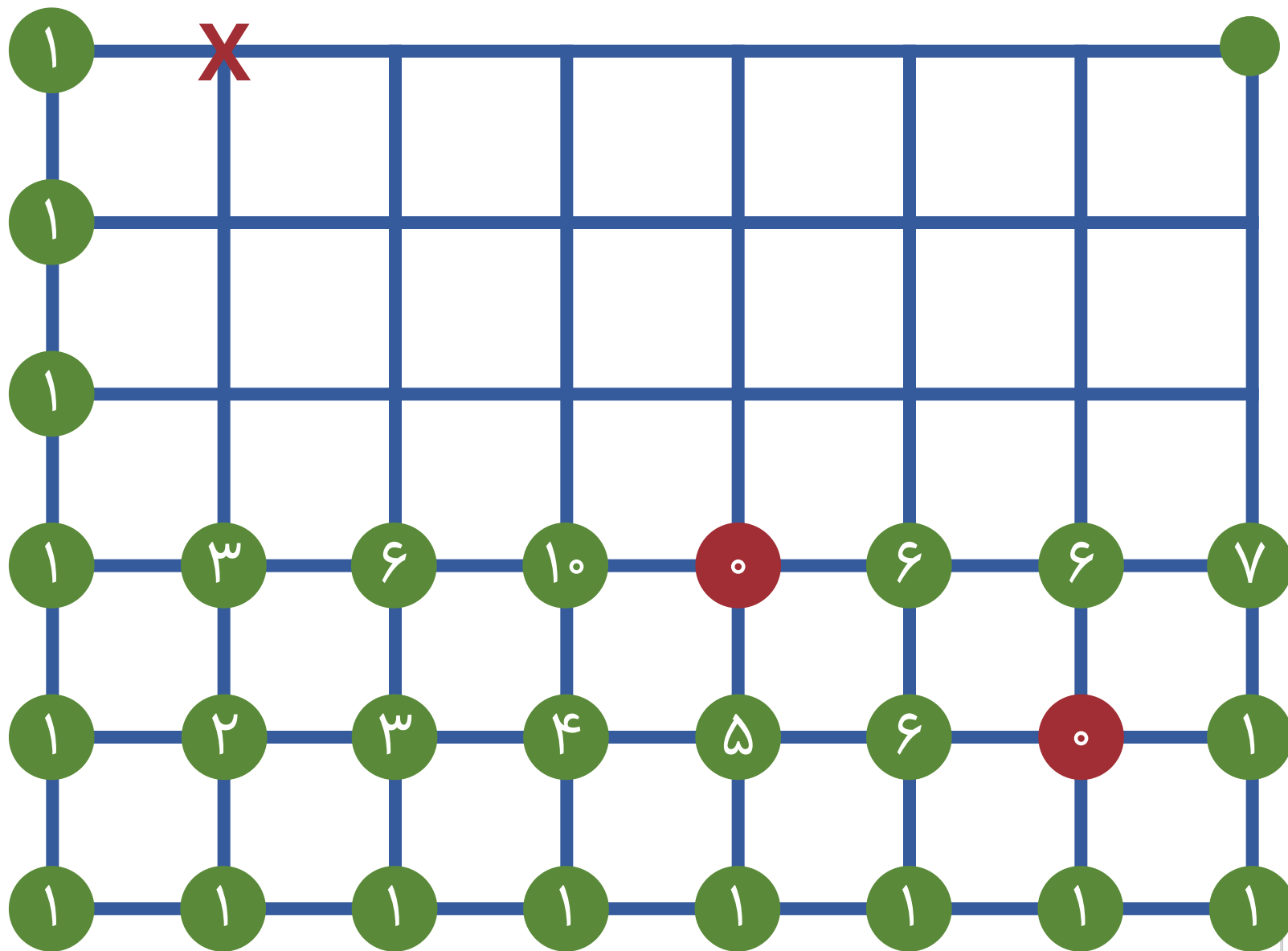
اگر بعضی تقاطعها مسدود باشد چطور تعداد مسیرها را بیابیم؟



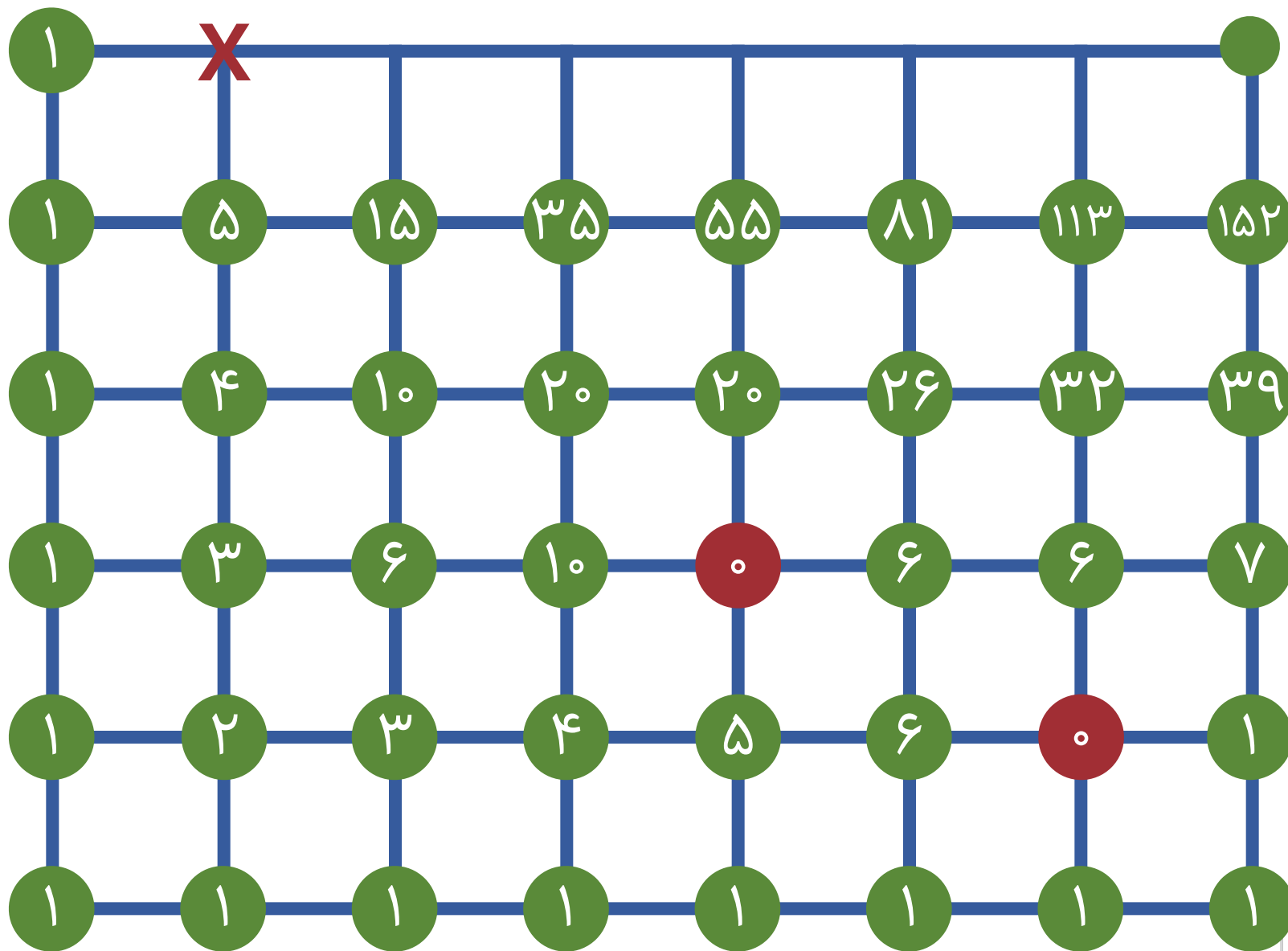
اگر بعضی تقاطعها مسدود باشد چطور تعداد مسیرها را بیابیم؟



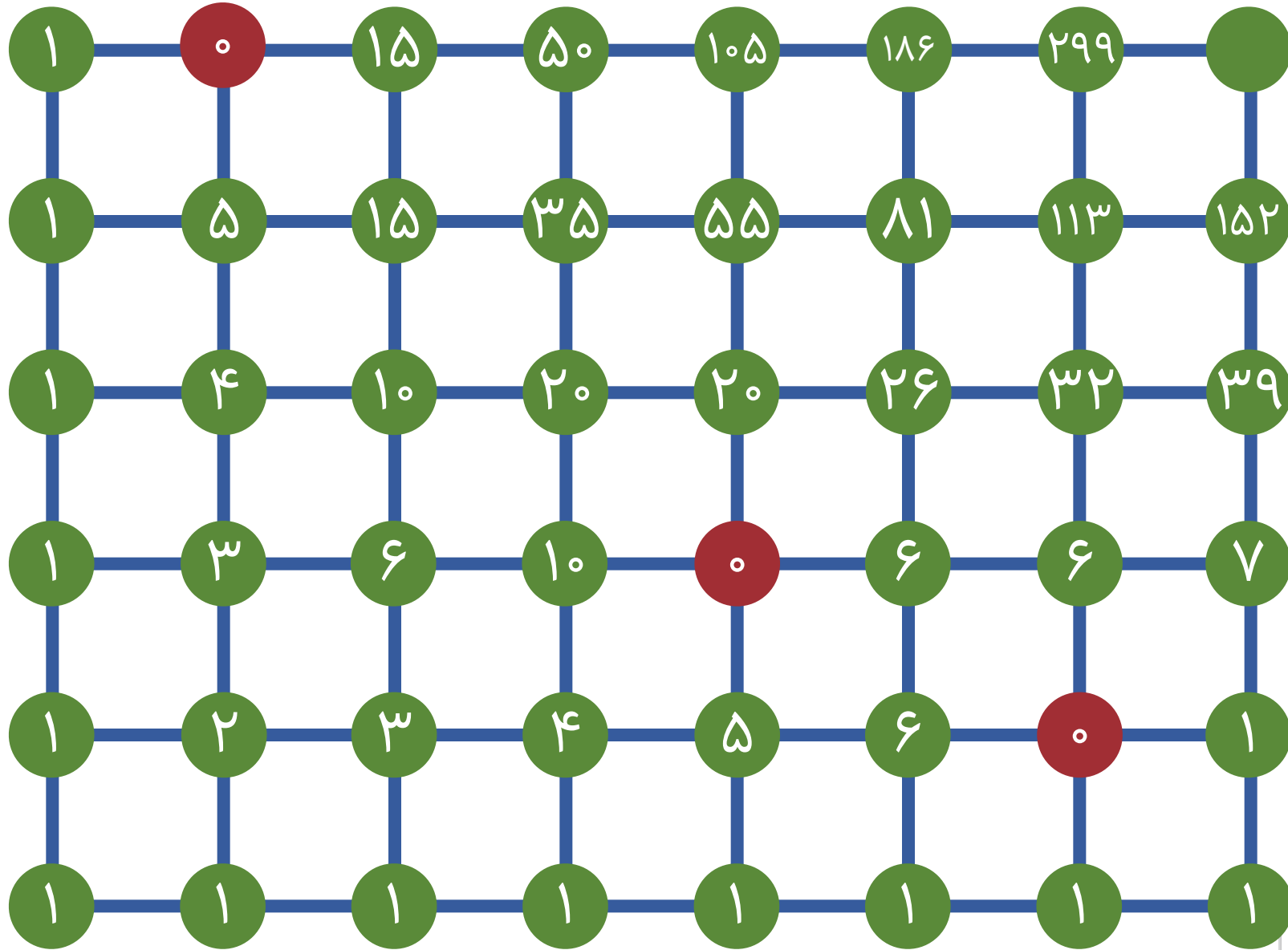
اگر بعضی تقاطعها مسدود باشد چطور تعداد مسیرها را بیابیم؟



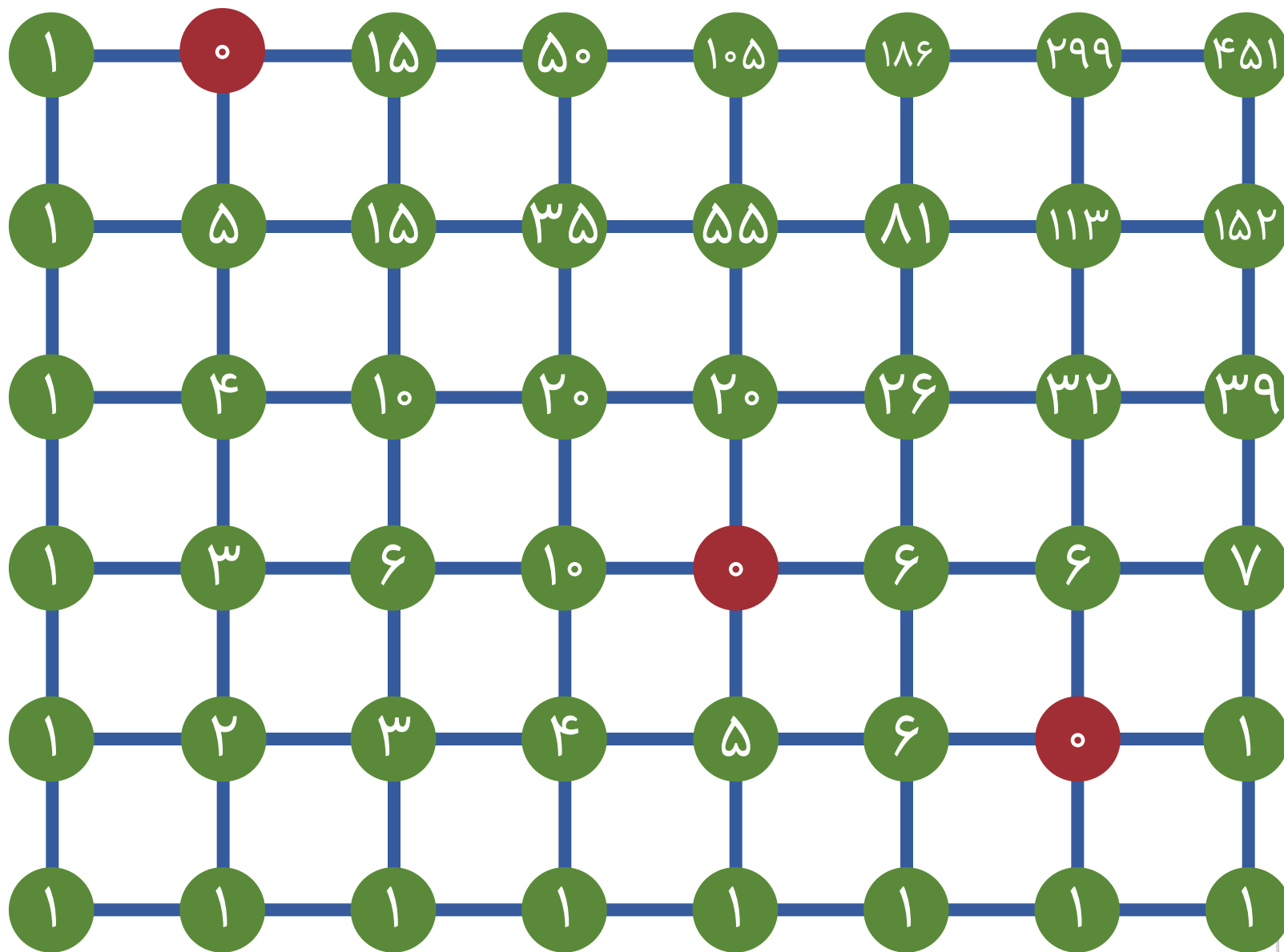
اگر بعضی تقاطعها مسدود باشد چطور تعداد مسیرها را بیابیم؟



اگر بعضی تقاطعها مسدود باشد چطور تعداد مسیرها را بیابیم؟



اگر بعضی تقاطعها مسدود باشد چطور تعداد مسیرها را بیابیم؟



الگوریتم یافتن تعداد مسیرها

• ابعاد صفحه m و n و همچنین وضعیت بلوک‌ها b را بخوان.



الگوریتم یافتن تعداد مسیرها

- ابعاد صفحه m و n و همچنین وضعیت بلوک‌ها b را بخوان.
- آرایه‌ی s را با ابعاد m در n و مقادیر اولیه‌ی صفر در نظر بگیر.



الگوریتم یافتن تعداد مسیرها

- ابعاد صفحه m و n و همچنین وضعیت بلوک‌ها b را بخوان.
- آرایه‌ی s را با ابعاد m در n و مقادیر اولیه‌ی صفر در نظر بگیر.
- $s[0][0]=1$

الگوریتم یافتن تعداد مسیرها

- ابعاد صفحه m و n و همچنین وضعیت بلوک‌ها b را بخوان.
 - آرایه‌ی s را با ابعاد m در n و مقادیر اولیه‌ی صفر در نظر بگیر.
 - $s[0][0]=1$
 - به ازای هر i بین 1 تا $m-1$:
- اگر $b[i][0]==1$ آنگاه $s[i][0] = s[i-1][0]$

الگوریتم یافتن تعداد مسیرها

- ابعاد صفحه m و n و همچنین وضعیت بلوک‌ها b را بخوان.
- آرایه‌ی s را با ابعاد m در n و مقادیر اولیه‌ی صفر در نظر بگیر.
- $s[0][0]=1$
- به ازای هر i بین 1 تا $m-1$:
اگر $b[i][0]==1$ آنگاه $s[i][0] = s[i-1][0]$
- به ازای هر j بین 1 تا $n-1$:
اگر $b[0][j]==1$ آنگاه $s[0][j] = s[0][j-1]$

الگوریتم یافتن تعداد مسیرها

- ابعاد صفحه m و n و همچنین وضعیت بلوک‌ها b را بخوان.
- آرایه‌ی s را با ابعاد m در n و مقادیر اولیه‌ی صفر در نظر بگیر.
- $s[0][0]=1$
- به ازای هر i بین 1 تا $m-1$:
اگر $b[i][0]==1$ آنگاه $s[i][0] = s[i-1][0]$
- به ازای هر j بین 1 تا $n-1$:
اگر $b[0][j]==1$ آنگاه $s[0][j] = s[0][j-1]$
- به ازای هر i بین 1 تا $m-1$:
به ازای هر j بین 1 تا $n-1$:

الگوریتم یافتن تعداد مسیرها

- ابعاد صفحه m و n و همچنین وضعیت بلوک‌ها b را بخوان.
- آرایه‌ی s را با ابعاد m در n و مقادیر اولیه‌ی صفر در نظر بگیر.

$$s[0][0]=1$$

- به ازای هر i بین 1 تا $m-1$:

$$s[i][0] = s[i-1][0] \text{ اگر } b[i][0]==1 \text{ آنگاه}$$

- به ازای هر j بین 1 تا $n-1$:

$$s[0][j] = s[0][j-1] \text{ اگر } b[0][j]==1 \text{ آنگاه}$$

- به ازای هر i بین 1 تا $m-1$:

به ازای هر j بین 1 تا $n-1$:

$$s[i][j] = s[i-1][j] + s[i][j-1] \text{ اگر } b[i][j]==1 \text{ آنگاه}$$

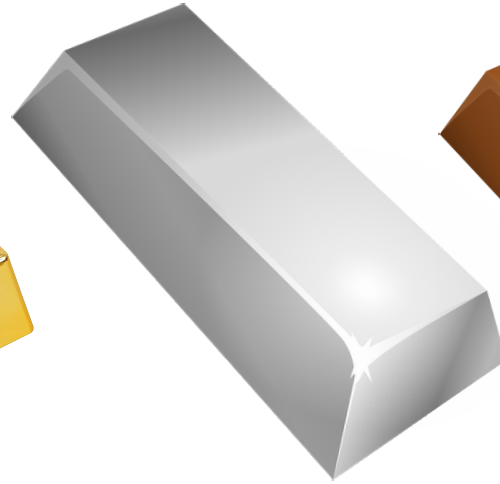


وقت كدزدن (:

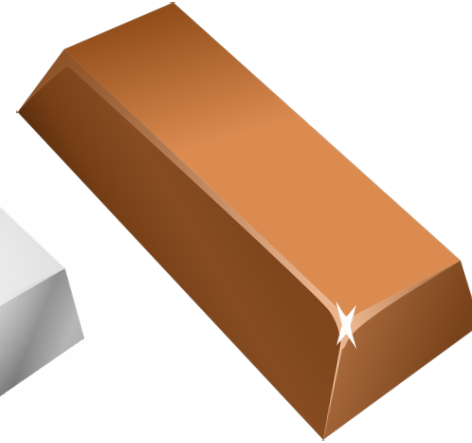
برگردیم به مسالهی کوله‌پشتی



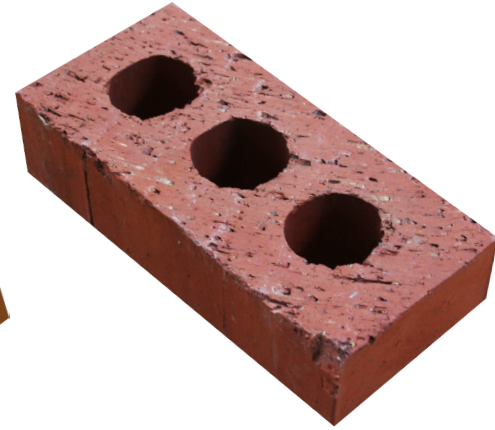
۱۰۰ ۱



۲۰ ۶



۱۰ ۳



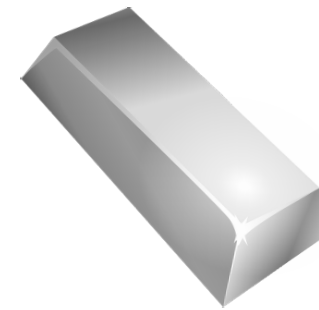
۱ ۱



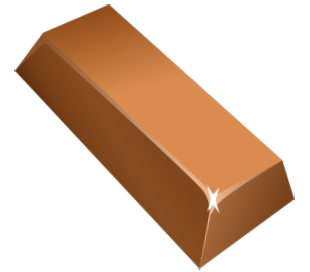
۱ ۲



۸۰ ۱



۱۰ ۲

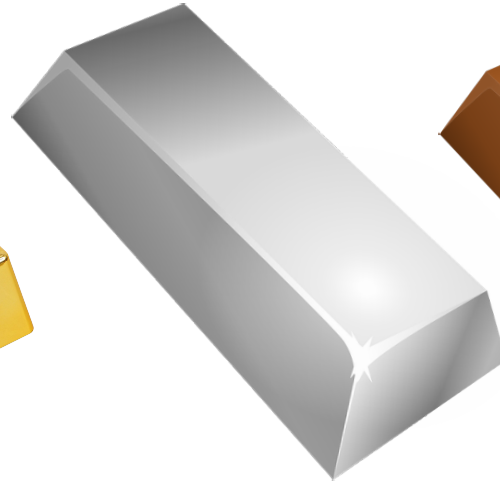


۵ ۲

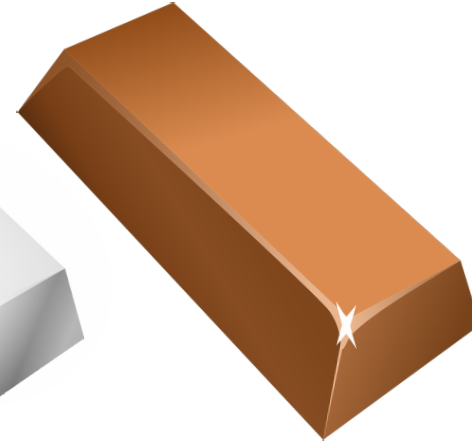
کالاها را شماره گذاری می کنیم.



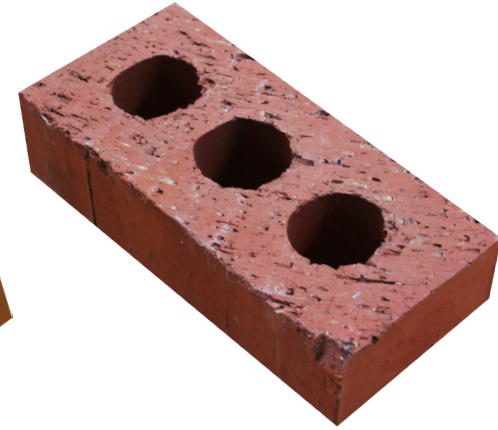
0



1



2



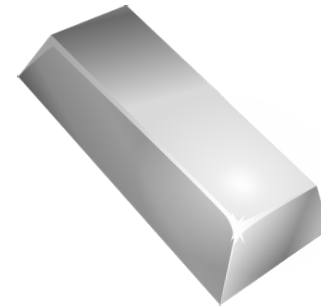
3



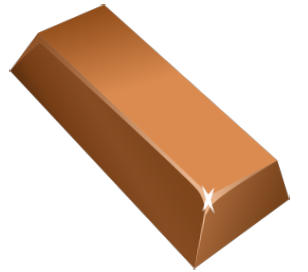
4



5

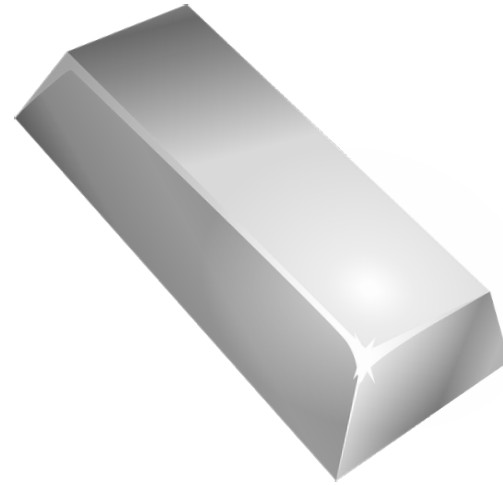


6



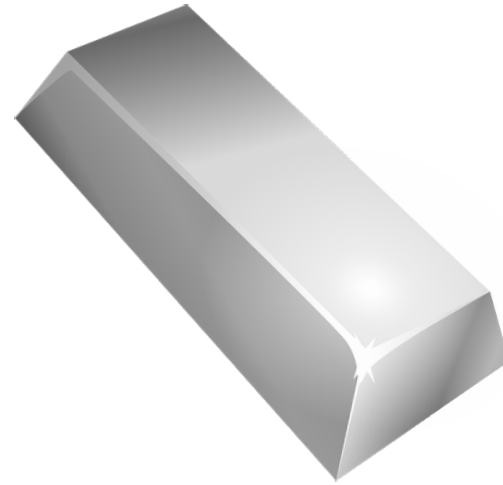
7

به ازای هر کالای i دو متغیر $p[i]$ (قیمت) و $w[i]$ (وزن) در نظر می‌گیریم.



۲۰ ۶

به ازای هر کالای i دو متغیر $p[i]$ (قیمت) و $w[i]$ (وزن) در نظر می‌گیریم.



۲۰ ۶

$$p[1] = 20$$

$$w[1] = 6$$

چه تعریفی را برای $s[i][j]$ در نظر بگیریم؟



چه تعریفی را برای $s[i][j]$ در نظر بگیریم؟

$s[i][j]$ برابر است با بیشترین مجموع قیمت کالاها که می‌توانیم در کوله‌پشتی قرار دهیم، اگر:

چه تعریفی را برای $s[i][j]$ در نظر بگیریم؟

$s[i][j]$ برابر است با بیشترین مجموع قیمت کالاها که می‌توانیم در کوله‌پشتی قرار دهیم، اگر:

- فقط حق برداشتن کالاهای 0 تا i را داشته باشیم.

چه تعریفی را برای $s[i][j]$ در نظر بگیریم؟

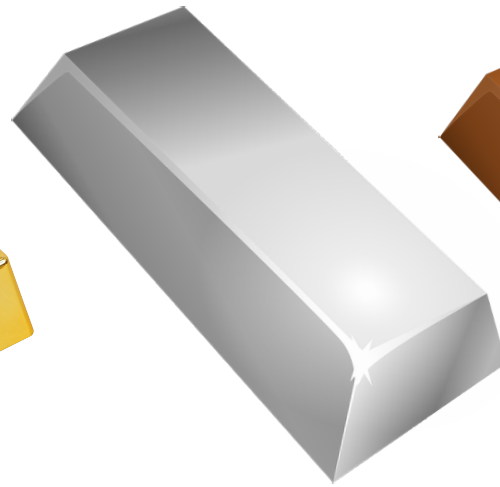
$s[i][j]$ برابر است با بیشترین مجموع قیمت کالاها که می‌توانیم در کوله‌پشتی قرار دهیم، اگر:

- فقط حق برداشتن کالاهای 0 تا i را داشته باشیم.
- ظرفیت کوله‌پشتی z باشد.

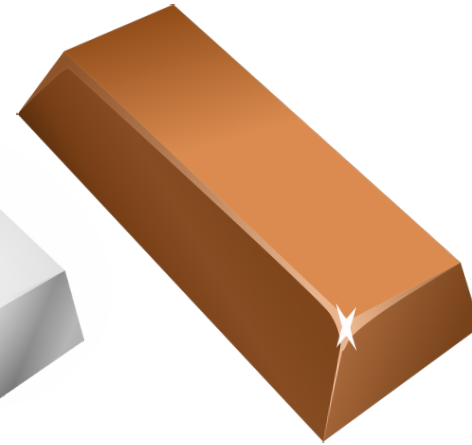
مثلا در حالتی که $i=2$ و $j=9$



۱۰۰ ۱



۲۰ ۶



۱۰ ۳



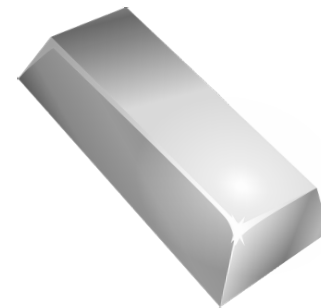
۱ ۱



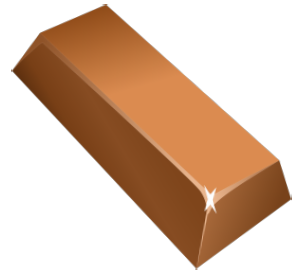
۱ ۲



۸۰ ۱



۱۰ ۲

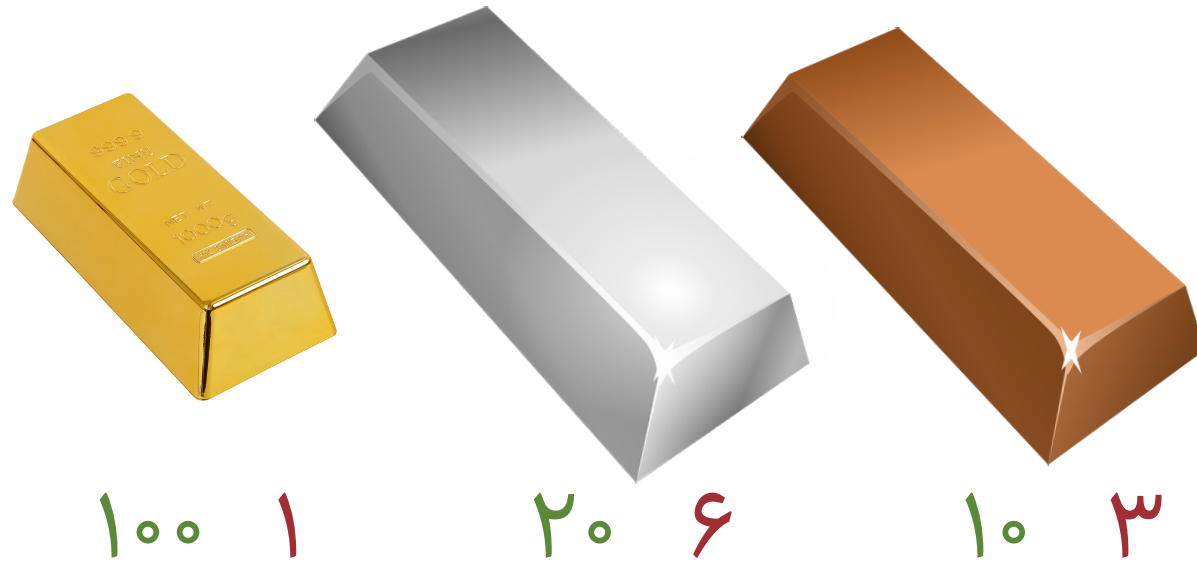


۵ ۲

مثلا در حالتی که $i=2$ و $j=9$



۹



۱۰۰ ۱

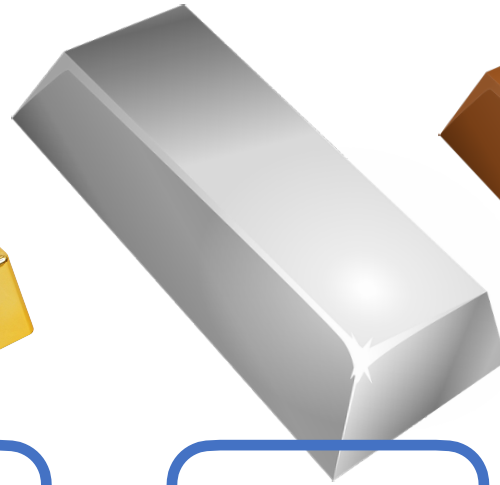
۲۰ ۶

۱۰ ۳

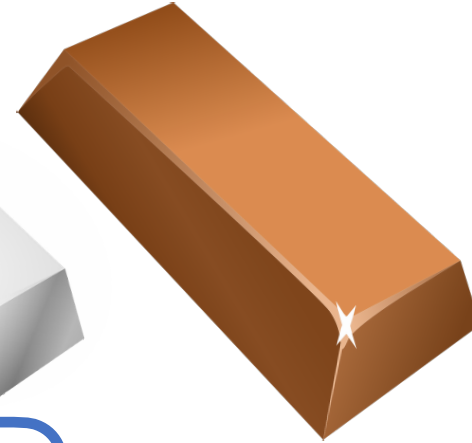
مثلا در حالتی که $i=2$ و $j=9$



۱۰۰ ۱



۲۰ ۶



۱۰ ۳

۱۲۰ ۹

چطور مقدار $s[i][j]$ را حساب کنیم؟



چطور مقدار $s[i][j]$ را حساب کنیم؟

• اگر کالای i ام را برنداریم، مثل آن که اصلا کالای i ام وجود نداشته:

چطور مقدار $s[i][j]$ را حساب کنیم؟

• اگر کالای i ام را برنداریم، مثل آن که اصلا کالای i ام وجود نداشته:

$$s[i][j] = s[i-1][j]$$

چطور مقدار $s[i][j]$ را حساب کنیم؟

• اگر کالای i ام را برداریم، مثل آن که اصلا کالای i ام وجود نداشته:

$$s[i][j] = s[i-1][j]$$

• اگر کالای i ام را برداریم، باید:

چطور مقدار $s[i][j]$ را حساب کنیم؟

• اگر کالای i ام را برداریم، مثل آن که اصلا کالای i ام وجود نداشته:

$$s[i][j] = s[i-1][j]$$

• اگر کالای i ام را برداریم، باید:

• وزنش را از ظرفیت کوله‌پشتی کم کنیم.

چطور مقدار $s[i][j]$ را حساب کنیم؟

• اگر کالای i ام را برداریم، مثل آن که اصلا کالای i ام وجود نداشته:

$$s[i][j] = s[i-1][j]$$

• اگر کالای i ام را برداریم، باید:

• وزنش را از ظرفیت کوله‌پشتی کم کنیم.

• قیمتش را به جمع قیمت کالاهایی که برداشته‌ایم بیافزاییم.

چطور مقدار $s[i][j]$ را حساب کنیم؟

• اگر کالای i ام را برداریم، مثل آن که اصلا کالای i ام وجود نداشته:

$$s[i][j] = s[i-1][j]$$

• اگر کالای i ام را برداریم، باید:

• وزنش را از ظرفیت کوله‌پشتی کم کنیم.

• قیمتش را به جمع قیمت کالاهایی که برداشته‌ایم بیافزاییم.

$$s[i][j] = s[i-1][j-w[i]] + p[i]$$

چطور مقدار $s[i][j]$ را حساب کنیم؟

• اگر کالای i ام را برداریم، مثل آن که اصلا کالای i ام وجود نداشته:

$$s[i][j] = s[i-1][j]$$

• اگر کالای i ام را برداریم، باید:

• وزنش را از ظرفیت کوله‌پشتی کم کنیم.

• قیمتش را به جمع قیمت کالاهایی که برداشته‌ایم بیافزاییم.

$$s[i][j] = s[i-1][j-w[i]] + p[i]$$

الگوریتم پرکردن کوله‌پشتی

• تعداد کالاها n ، ظرفیت کوله‌پشتی m و قیمت $p[i]$ و وزن $w[i]$ هر کالا را بخوان.



الگوریتم پرکردن کوله‌پشتی

- تعداد کالاها n ، ظرفیت کوله‌پشتی m و قیمت $p[i]$ و وزن $w[i]$ هر کالا را بخوان.
- آرایه‌ی s را با ابعاد n در $m+1$ و مقادیر اولیه‌ی صفر در نظر بگیر.



الگوریتم پرکردن کوله‌پشتی

- تعداد کالاها n ، ظرفیت کوله‌پشتی m و قیمت $p[i]$ و وزن $w[i]$ هر کالا را بخوان.
- آرایه‌ی s را با ابعاد n در $m+1$ و مقادیر اولیه‌ی صفر در نظر بگیر.
- به ازای هر j بین $w[0]$ تا m
 $s[0][j]=p[0]$



الگوریتم پرکردن کوله‌پشتی

- تعداد کالاها n ، ظرفیت کوله‌پشتی m و قیمت $p[i]$ و وزن $w[i]$ هر کالا را بخوان.
- آرایه‌ی s را با ابعاد n در $m+1$ و مقادیر اولیه‌ی صفر در نظر بگیر.
- به ازای هر j بین $w[0]$ تا m
 - $s[0][j]=p[0]$
- به ازای هر i بین 1 تا $n-1$
 - به ازای هر j بین 1 تا m

الگوریتم پرکردن کوله‌پشتی

- تعداد کالاها n ، ظرفیت کوله‌پشتی m و قیمت $p[i]$ و وزن $w[i]$ هر کالا را بخوان.
- آرایه‌ی s را با ابعاد n در $m+1$ و مقادیر اولیه‌ی صفر در نظر بگیر.

- به ازای هر j بین $w[0]$ تا m

$$s[0][j]=p[0]$$

- به ازای هر i بین 1 تا $n-1$

- به ازای هر j بین 1 تا m

$$s[i][j]=s[i-1][j]$$

الگوریتم پرکردن کوله‌پشتی

- تعداد کالاها n ، ظرفیت کوله‌پشتی m و قیمت $p[i]$ و وزن $w[i]$ هر کالا را بخوان.
- آرایه‌ی s را با ابعاد n در $m+1$ و مقادیر اولیه‌ی صفر در نظر بگیر.

- به ازای هر j بین $w[0]$ تا m

$$s[0][j]=p[0]$$

- به ازای هر i بین 1 تا $n-1$

- به ازای هر j بین 1 تا m

$$s[i][j]=s[i-1][j]$$

$$\text{اگر } w[i] \leq j$$

$$s[i][j]=\max(s[i][j], s[i-1][j-w[i]]+p[i])$$

وقت كدزدن (:

ثبت نام/ورود

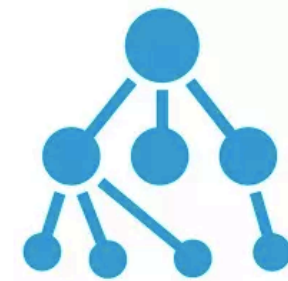


مرکز آموزش‌های الکترونیکی



خانه > لیست دوره‌ها > داده ساختارها و مبانی الگوریتم‌ها

ویدئوها
معرفی درس
مقدمه درس و آشنایی با پایتون
الگوریتم‌های ساده مرتب‌سازی
پیاده‌سازی و بررسی الگوریتم‌های مرتب‌سازی
انواع مرتب‌سازی: شمارشی و سطلی
مقایسه الگوریتم‌های مرتب‌سازی
پیچیدگی زمان الگوریتم‌ها
تحلیل الگوریتم‌های بازگشتی
تحلیل الگوریتم‌های بازگشتی (ادامه)
داده‌ساختارهای ساده: پشته، صف و لیست پیوندی



درباره دوره

مدرس

محتوای درس

پرسش و پاسخ (به زودی)

ثبت نام در دوره



در صورت بروز هرگونه مشکل از طریق فرم تماس با ما با ما در ارتباط باشید.



ثبت نام/ورود

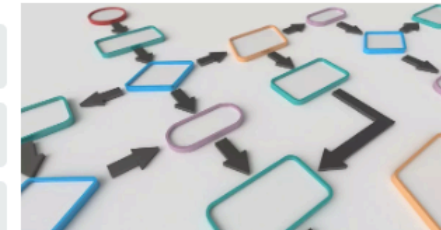


مرکز آموزش‌های الکترونیکی



خانه < لیست دوره‌ها < آموزش طراحی الگوریتم‌ها

<	تمرین‌ها
<	یادداشت‌ها
∨	ویدئوها
📄	مقدمه - الگوریتم‌های حریمانه
📄	الگوریتم‌های حریمانه - مسئله زمان‌بندی فعالیت‌ها
📄	الگوریتم‌های حریمانه - فشرده‌سازی و کد هافمن
📄	حریمانه و برنامه‌نویسی پویا، مسئله‌ی کوله‌پشتی
📄	برنامه‌نویسی پویا - مسئله‌ی تبدیل رشته‌ها
📄	برنامه‌نویسی پویا - ضرب ماتریس، بزرگترین زیردنباله‌ی صعودی، بزرگترین زیردنباله‌ی مشترک
📄	برنامه‌نویسی پویا - مجموعه‌ی مستقل در درخت
📄	درخت پوشای کمینه (Minimum Spanning Tree)



درباره دوره 📄

مدرس 👤

محتوای درس 📄

پرسش‌وپاسخ (به زودی) ❓

ثبت نام در دوره



در صورت بروز هرگونه مشکل از طریق فرم تماس با ما با ما در ارتباط باشید.



عضویت

ورود

ورود به بخش شرکتها

بوتکمپ 🔥

جونپورا

کاریابی

مسابقات

بانک سؤالات

کالج

سامانه آموزشی

Quera

خانه < مسابقات

برترینها ★

مسابقات منتشرنشده

مسابقات خصوصی

مسابقات دیگر

1
مسابقات ویژه

مسابقات 📄

رتبه	نام	امتیاز
۱	علی شاه علی	۲۴۶۸
۲	علی صفری	۲۴۵۱
۳	Mike	۲۴۰۲
۴	ابوالفضل سلطانی	۲۳۳۳
۵	شایان پردیس	۲۳۳۲
۶	شایان چشم جهان	۲۳۳۰
۷	امیررضا پوراخوان	۲۲۳۴
۸	محمد صانعیان	۲۲۳۲
۹	علی پاشا منتصری	۲۲۰۸
۱۰	Mehrdad Saberi	۲۱۹۶
🏆 رتبه‌بندی کامل		

+ درخواست ایجاد مسابقه

مسابقات در حال اجرا

نام مسابقه	زمان تا شروع	مدت مسابقه	#سؤالات	#شرکت‌کنندگان
مسابقه الگوریتمی تابستون چلنج Quera	۱۶:۰۵ ۱۴۰۲/۶/۱۰ ۶ روز دیگر 📅	۲ دقیقه ساعت	۶	۱۰۹۱

مسابقات گذشته

نام مسابقه	زمان شروع	مدت مسابقه	#سؤالات	#شرکت‌کنندگان
مسابقه کریپتوکد (Python/Django) صرافی ارز دیجیتال تبدیل	۱۶:۰۵ ۱۴۰۲/۵/۱۳	۲ دقیقه ساعت	۵	۹۶۶

مسابقه الگوریتمی پردیس‌کد Quera	۱۶:۰۵ ۱۴۰۲/۴/۹	۲ دقیقه ساعت	۶	۱۴۵۸
---------------------------------	------------------	-----------------	---	------

مسابقه بله - مرحله دوم (Back-End)		۳	۳	
-----------------------------------	--	---	---	--



عضویت

ورود

ورود به بخش شرکت‌ها



بوت‌کمپ

کارایی

مسابقات

بانک سؤالات

کالج

سامانه آموزشی

Quera

بانک سؤالات </> نفرات برتر

بانک سؤالات



بیشترین حل



جستجوی نام یا شماره سؤال...



نشان‌شده



وضعیت حل

- حل کامل
- حل ناتمام
- بدون تلاش



دسته‌بندی

- تکنولوژی
- مسابقه
- دانشگاهی
- المپیاد



سختی جدید

- ساده
- متوسط
- سخت

نام سؤال	منبع سؤال	تعداد حل	سختی	برچسب‌ها
سؤال زرد	کدکاپ ۲، مسابقه مقدماتی	۱۲۳۸۵ از ۱۳۵۱۷	ساده	پیاده سازی
یک سؤال ساده	مسابقه شماره ۱۰ (سطح ۲)	۱۰۷۸۳ از ۱۱۶۰۱	ساده	پیاده سازی
فاکتوریل	-	۹۴۱۷ از ۹۹۱۰	ساده	مبانی برنامه‌نویسی، دانشگاه صنعتی شریف
یخدارچی	کدکاپ ۲، مسابقه تمرینی ۲	۹۱۲۲ از ۹۸۴۶	ساده	پیاده سازی
مشق امشب باقر	کدکاپ ۳، مقدماتی	۸۳۸۹ از ۹۹۲۰	ساده	ریاضیات، هندسه
اعداد فیثاغورثی	-	۸۱۳۶ از ۹۲۰۱	ساده	مبانی برنامه‌نویسی، دانشگاه صنعتی شریف
تک‌رقمی	کدکاپ ۲، مسابقه مقدماتی	۷۶۹۸ از ۹۵۲۹	ساده	ریاضیات
جدول ضرب گنده	کدکاپ ۲، مسابقه آموزشی ۲	۷۵۴۷ از ۷۹۱۴	ساده	پیاده سازی



https://opedia.ir

ورود به سیستم

المپدیا
دانشنامه‌ی المپیاد کامپیوتر ایران

محل شما: فهرست

فهرست

فهرست

این وبگاه شامل مطالب آموزشی، سوالات و اطلاعات مربوط به مراحل مختلف المپیاد کامپیوتر ایران است.

درباره‌ی المپیاد

- دوره‌های المپیاد
- نتایج ایران در مسابقات جهانی
- اعضای کمیته‌ی ملی
- همکاران کمیته‌ی ملی
- دانش‌پژوهان
- سایت‌های مفید

سوالات المپیاد

- مرحله‌ی اول
- مرحله‌ی دوم
- مرحله‌ی سوم (برنامه‌نویسی)
- دوره‌ی تابستان
- دوره‌ی انتخاب تیم
- سوالات متفرقه
- بایگانی آزمون‌ها

آموزش المپیاد

- ترکیبیات
- گراف
- الگوریتم
- برنامه‌نویسی
- الگوریتم‌های تکمیلی
- آماده‌سازی برای المپیاد
- سایت‌های مفید

خبرگاه المپیاد کامپیوتر

منبع رسمی اخبار و اطلاعیه‌های کمیته ملی المپیاد کامپیوتر ایران



خانه | المپدیا: دانش‌نامه‌ی المپیاد کامپیوتر ایران | اعضای کمیته ملی | تماس با ما

افتخار آفرینی تیم المپیاد کامپیوتر ایران

۷ دیدگاه‌نوشته شده توسط مدیر, شهریور ۲۱, ۱۳۹۹, ۲:۴۰ ب.ظ

تیم المپیاد کامپیوتر ایران با کسب ۳ مدال طلا و یک مدال نقره موفق شد بطور مشترک با دو تیم روسیه و آمریکا بعد از چین در جایگاه دوم به لحاظ مدالی قرار گیرد که بهترین نتیجه مدالی در تمام ادوار حضور تیم ایران در مسابقات جهانی است. همچنین در رتبه‌بندی نمره‌ای ایران بعد از چین، آمریکا و کره در جایگاه ۴ام قرار گرفت. رتبه ایران در تمام ادوار حضور ایران در مسابقات جهانی را می‌توانید در اینجا مشاهده کنید. اعضای تیم به همراه مدال آن‌ها در زیر آمده است.

- شایان پردیس (مدال طلا)
- کسری مظاهری (مدال طلا)
- علی صفری (مدال طلا)
- ابوالفضل سلطانی (مدال نقره)

به همه این عزیزان و خانواده‌های گرامی‌شان صمیمانه تبریک می‌گوییم.

برای تماس با کمیته ملی المپیاد کامپیوتر، نامه‌های خود را به نشانی inoi[at]inoi.ir ارسال نمائید، یا در صفحه‌ی تماس با ما پیغام بگذارید.

پیوندها

- المپدیا: دانش‌نامه‌ی المپیاد کامپیوتر ایران
- المپیاد جهانی کامپیوتر
- المپیاد جهانی کامپیوتر ۲۰۱۷
- باشگاه دانش‌پژوهان جوان
- کانال تلگرامی کمیته ملی کامپیوتر
- کاهو: سایت پرسش و پاسخ المپیاد کامپیوتر
- گروه inoi-news برای دریافت ایمیلی اخبار سایت
- وبگاه المپیاد ریاضی ایران



Harbour.Space Scholarship Contest 2023-2024 (Div. 1 + Div. 2)

By [Parisa_Amiri](#), 7 days ago, 

Salam, Codeforces!

We're so excited to invite you to take part in our round [Harbour.Space Scholarship Contest 2023-2024 \(Div. 1 + Div. 2\)](#) which will start on [Saturday, August 26, 2023 at 18:05^{UTC+3.5}](#). The round will be **rated and open for everyone**.

The problems were prepared and authored by [amenotiomoi](#), [Psychotic_D](#), [wuhudsm](#), [Parisa_Amiri](#), [chromate00](#), [JohnVictor](#), [ODT](#), [ugly2333](#), [Lavine](#), [RiverHamster](#), [flowerletter](#) and [AquaMoon](#).

We would like to thank :

- [Irkstepanov](#) for coordinating the round;
- [q-w-q-w-q](#), [BurnedChicken](#), [p_b_p_b](#) and [Heltion](#) for **LGM** testing;
- [hibye1217](#), [alireza_kaviani](#), [errorgorn](#), [DataStructures](#), [AmShZ](#), [Crying](#), [Huah](#), [bthero](#), [fengzhengwei](#) and [Ecrade_](#) for **GM** and **IGM** testing;
- [Kieray](#), [-kiritto-](#), [Arpa](#), [triple__a](#), [blxst](#), [leukocyte](#), [ITO](#) and [NemanjaSo2005](#) for **M** and **IM** testing;
- [moonpay](#), [TAIYANGFENG](#), [callmepandey](#), [Mkswil](#) and [uuku](#) for **CM** testing;
- [bogocubic1](#), [KitasCraft](#), [aayushdhankecha](#), [Qualified](#), [Nanani](#), [Kinan](#) and [tilwtg](#) for **Expert** testing;
- [Hex-willappear](#), [Savani004](#), [AkibAzmain](#), [Vlsrut_](#), [s010456](#) and [Harshil4442](#) for **Specialist** testing;
- [strange-creator](#) for the only **Pupil** testing;
- [Valenz](#) for the only **Newbie** testing;
- [MikeMirzayanov](#) for the great Codeforces and Polygon platforms;
- And **at last but not least**, [You](#) for participating in the round!

→ Pay attention

Before contest

[Harbour.Space Scholarship Contest 2023-2024 \(Div. 1 + Div. 2\)](#)

07:30:11

[Register now »](#)

*has extra registration

→ Streams

[Upsolving EJOI? I can also debug your solutions? idk yet.](#)

By [kostka](#)

Before stream 10:55:09

[View all →](#)

→ Top rated

#	User	Rating
1	tourist	3727
2	cnnfls_csy	3621
3	Radewoosh	3598
4	-0.5	3545
5	jiangly	3514
6	amiya	3491
7	inaFSTream	3477



سایر سایت‌های یادگیری آنلاین

• مکتب‌خونه

• Coursera

• edX



Career Paths

Web Development Bootcamp

- Junior developer
- Full-stack developer
- Front-end developer
- Back-end developer
- iOS developer
- UI developer
- UX developer
- Data engineer
- Database analyst
- DevOps engineer
- Technical product manager

Data Science Bootcamp

- Data analyst
- Senior data analyst
- Junior data scientist
- Data scientist
- Senior data scientist
- Data systems developer
- Machine learning engineer
- DataOps engineer
- Business intelligence developer
- Data systems analyst

شاد و تن درست باشید :-)