

به نام داناترین

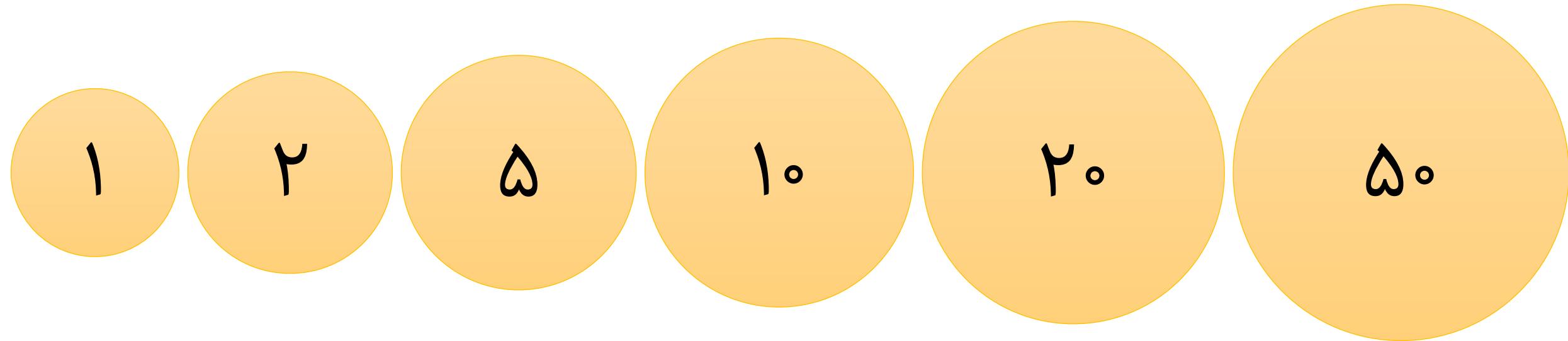


دورهی خلاقیت الگوریتمی و برنامهنویسی پایتون

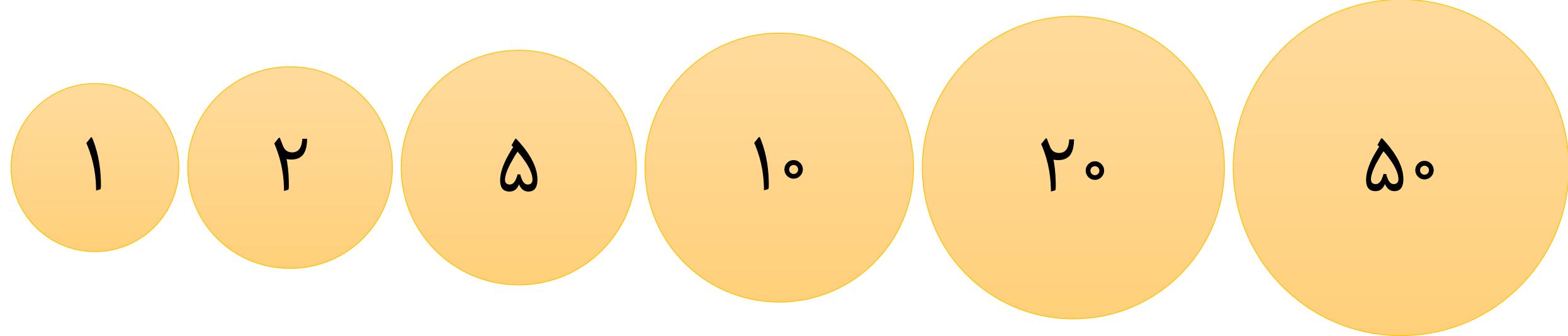
زندگی حرصانه!

دانشکدهی مهندسی کامپیوتر دانشگاه صنعتی شریف
تابستان ۱۴۰۲

زیل فروشنده است. او تعداد زیادی سکه ۱، ۵، ۱۰، ۲۰ و ۵۰ تومانی دارد.



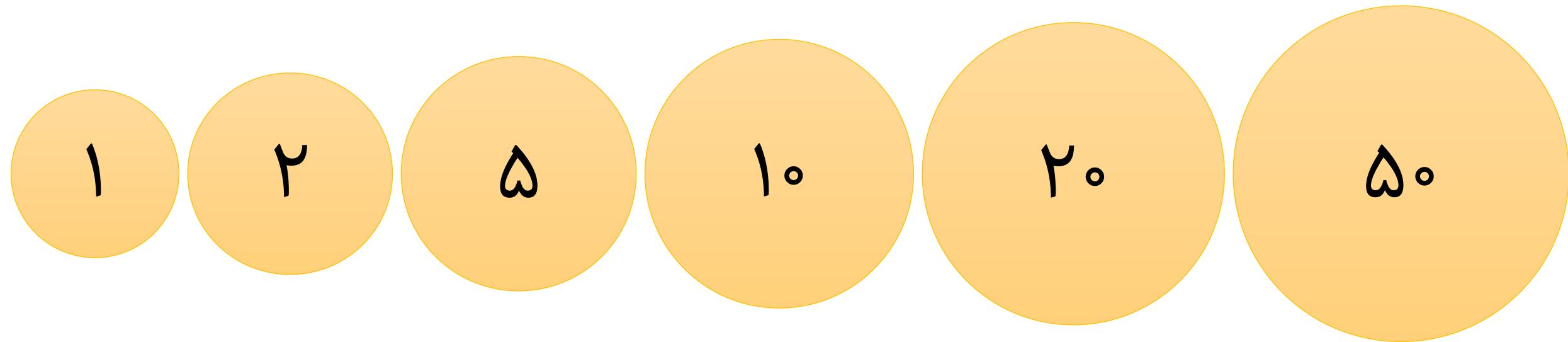
باقیماندهی پول هر مشتری را با این سکه‌ها پرداخت می‌کند.



باقیماندهی پول مشتری

۱۶

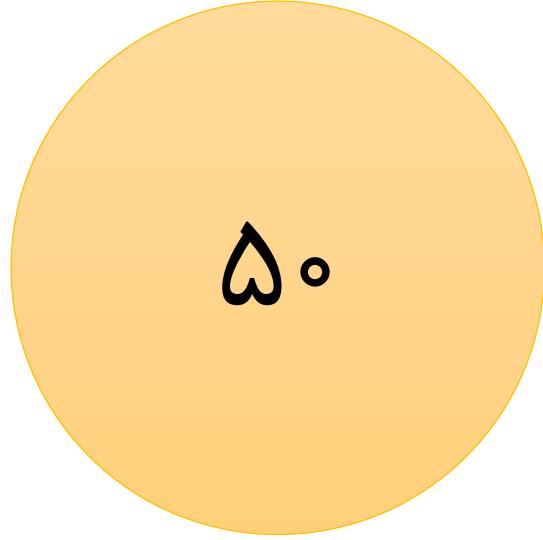
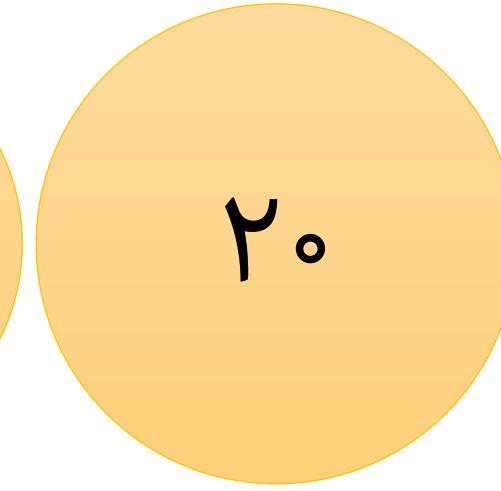
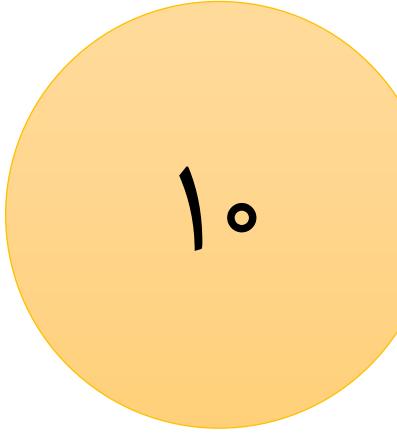
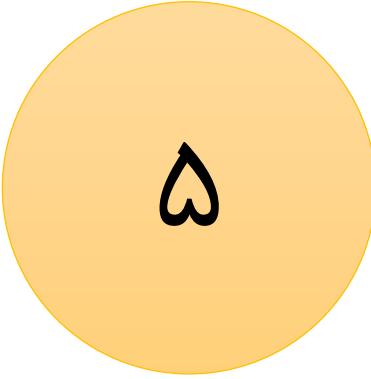
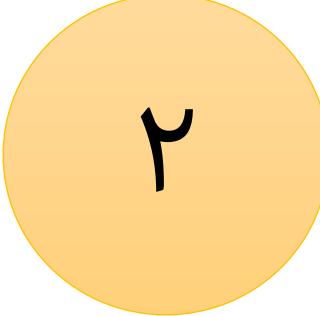
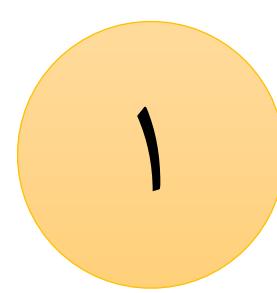
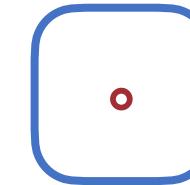
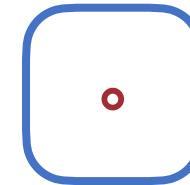
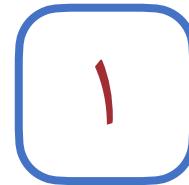
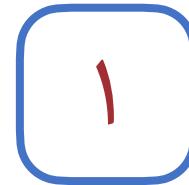
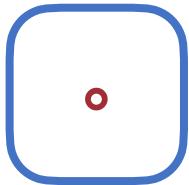
او می‌خواهد کمترین تعداد سکه را برای دادن باقیمانده استفاده کند.



باقیماندهی پول مشتری

۱۶

اگر باقیماندهی پول مشتری ۱۶ تومان باشد او به ۳ سکه نیاز دارد.



باقیماندهی پول مشتری



برای این مشتری حداقل تعداد سکه چیست؟

?

?

?

?

?

?

۱

۲

۵

۱۰

۲۰

۵۰

باقیماندهی پول مشتری

۳۴

برای این مشتری حداقل تعداد سکه چیست؟

۰

۲

۰

۱

۱

۰

۱

۲

۵

۱۰

۲۰

۵۰

باقیماندهی پول مشتری

۳۴

برای این مشتری حداقل تعداد سکه چیست؟

?

?

?

?

?

?

۱

۲

۵

۱۰

۲۰

۵۰

باقیماندهی پول مشتری

۱۴۷

برای این مشتری حداقل تعداد سکه چیست؟

۰

۱

۱

۰

۲

۲

۱

۲

۵

۱۰

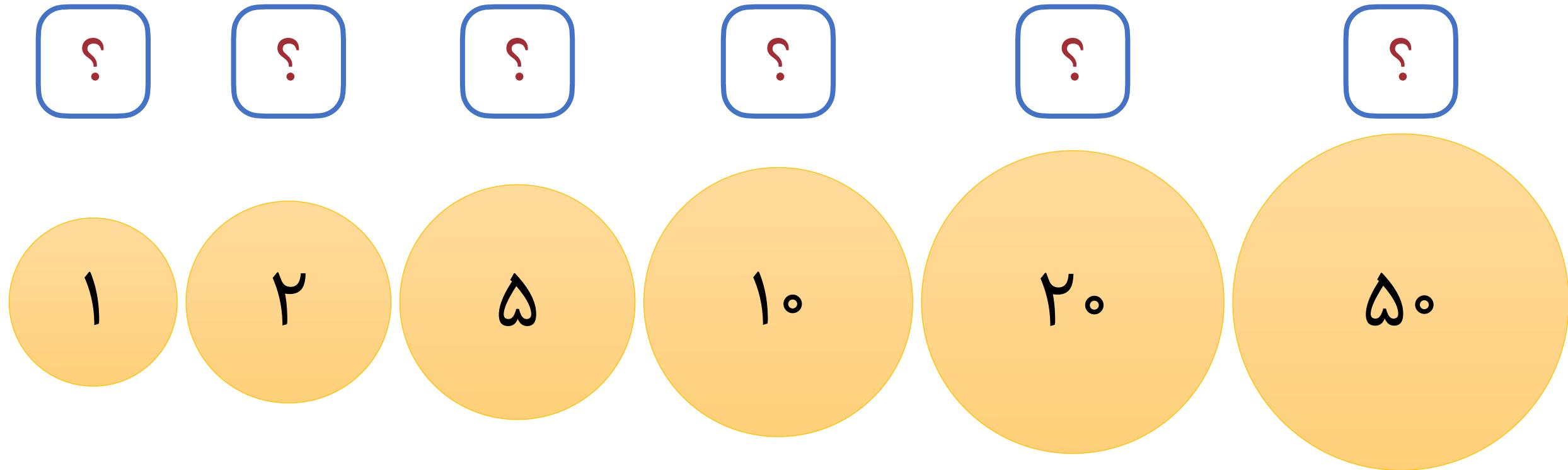
۲۰

۵۰

باقیماندهی پول مشتری

۱۴۷

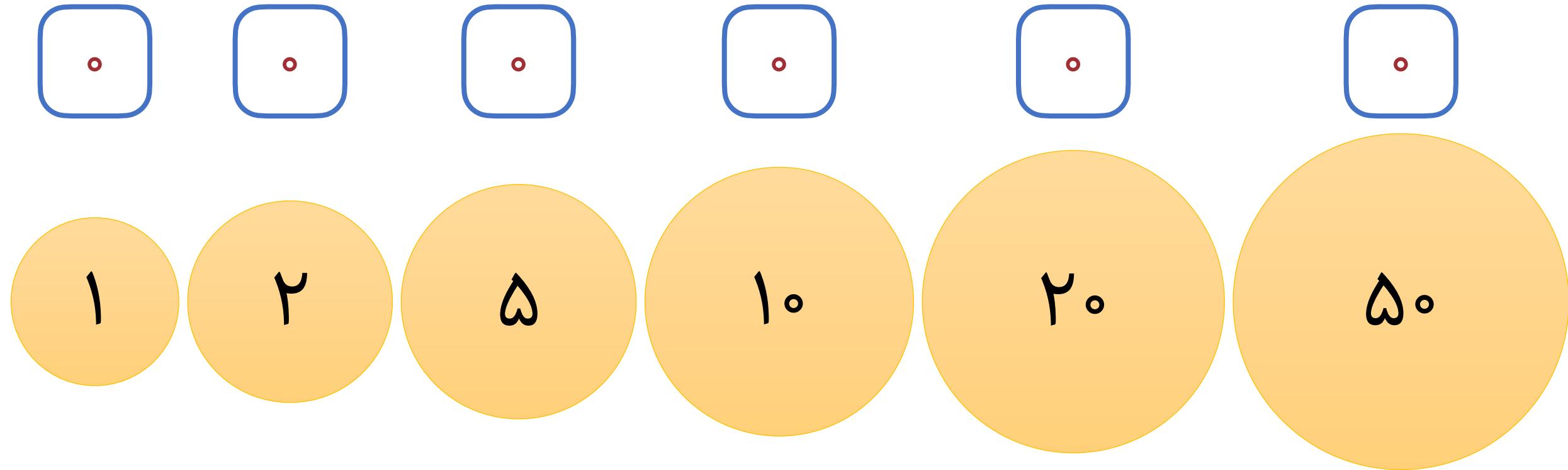
باید یک الگوریتم کشف کنیم!



باقیماندهی پول مشتری

۱۴۷

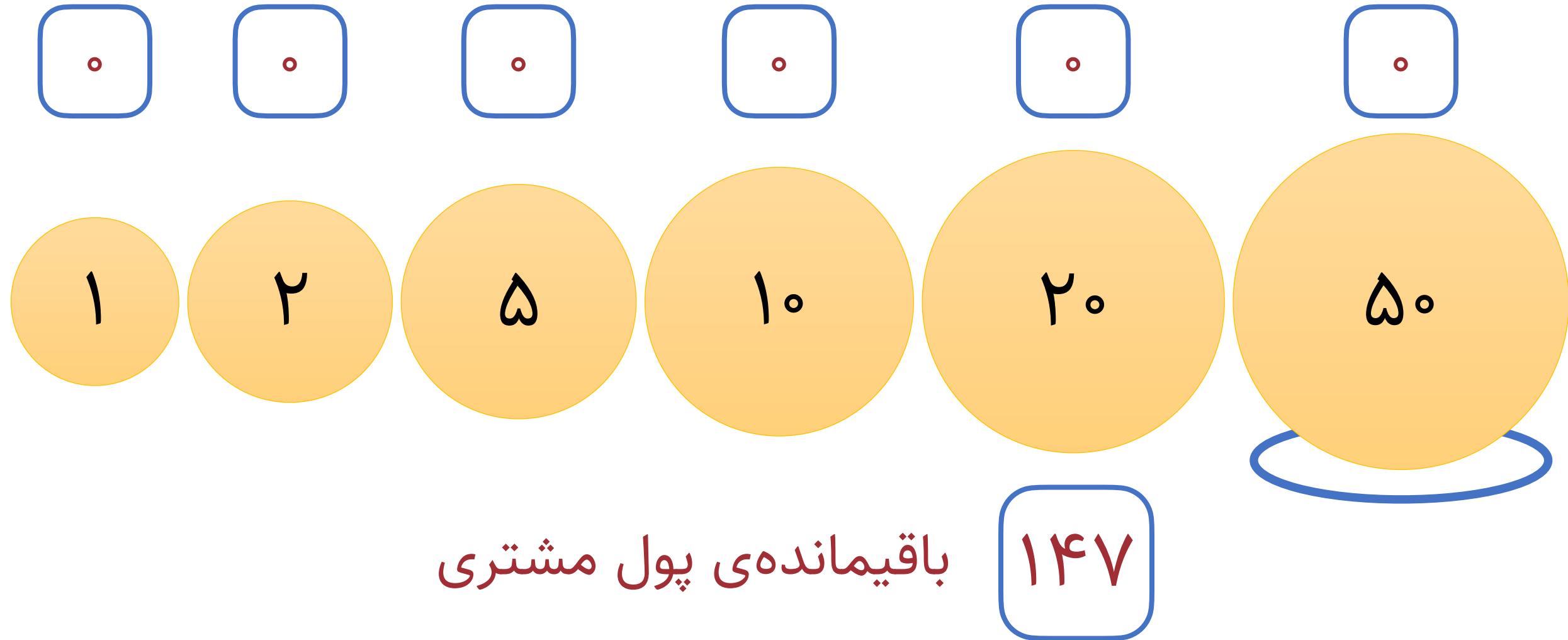
بیاید یک الگوریتم کشف کنیم!



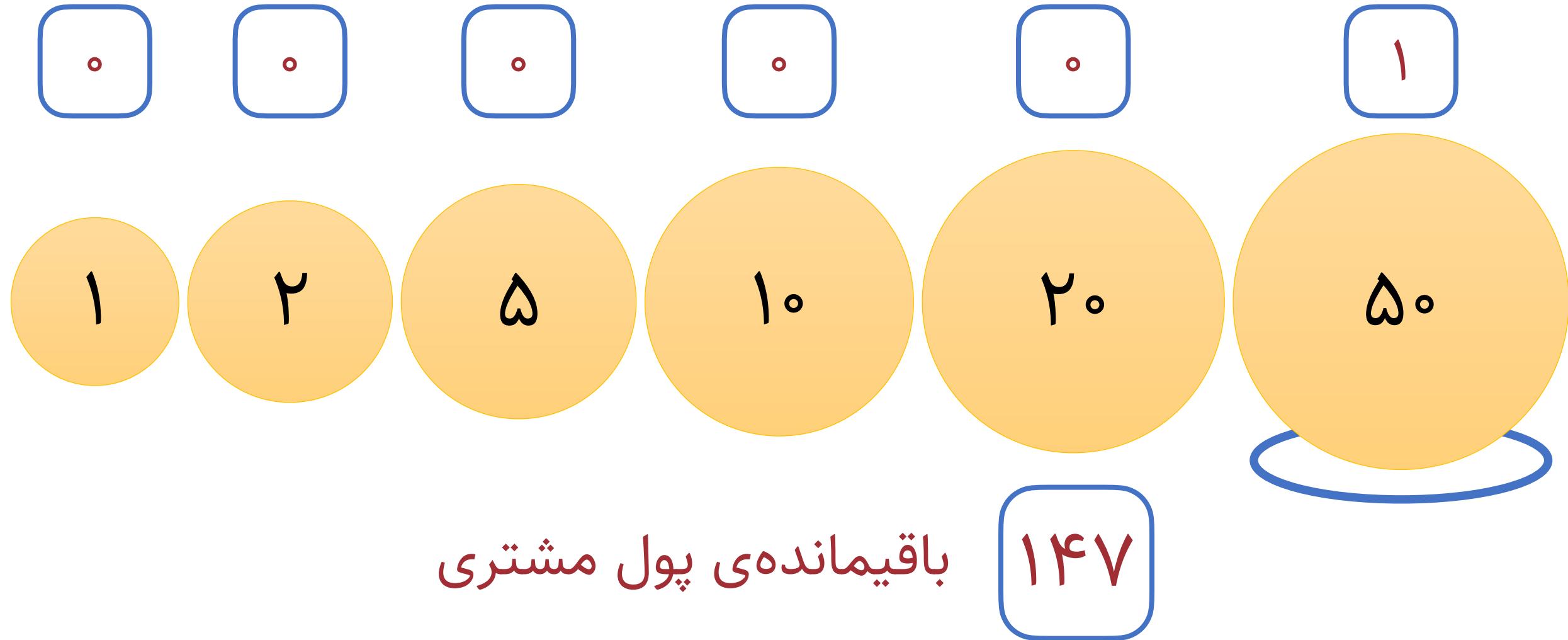
باقیماندهی پول مشتری

۱۴۷

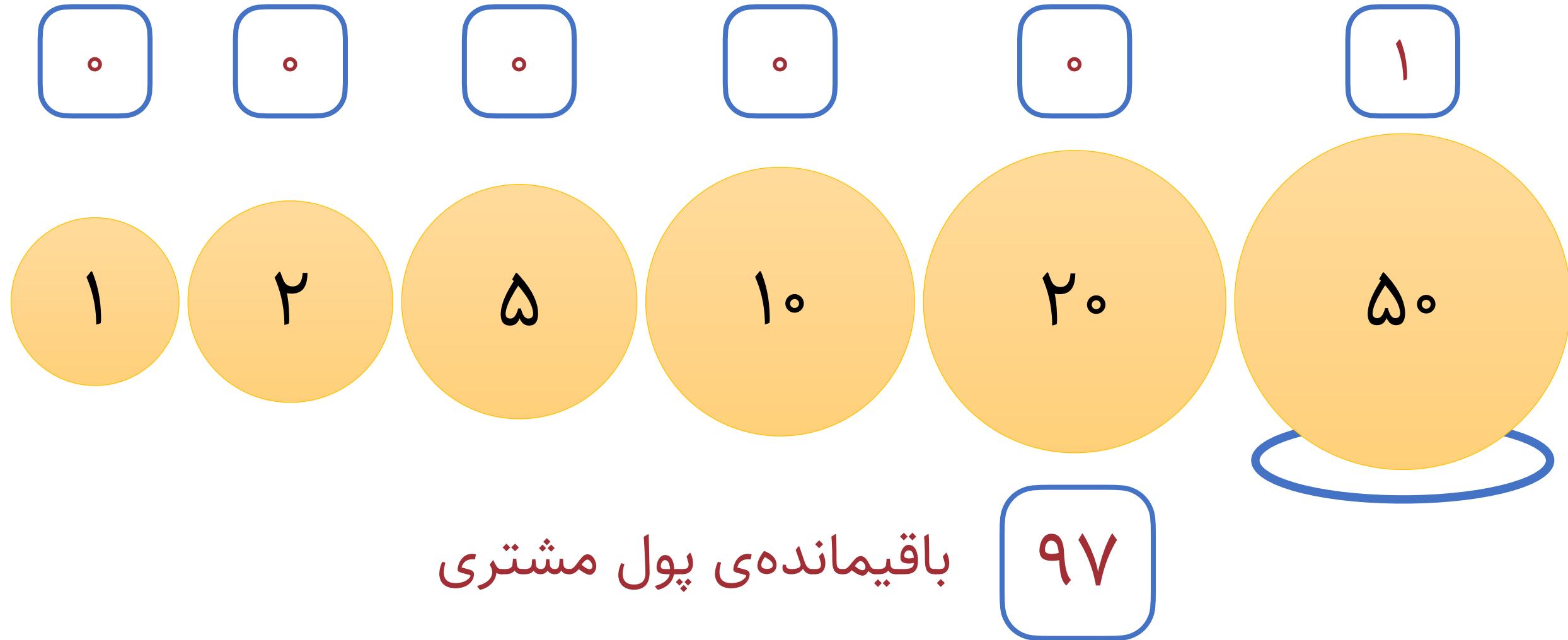
بیاید یک الگوریتم کشف کنیم!



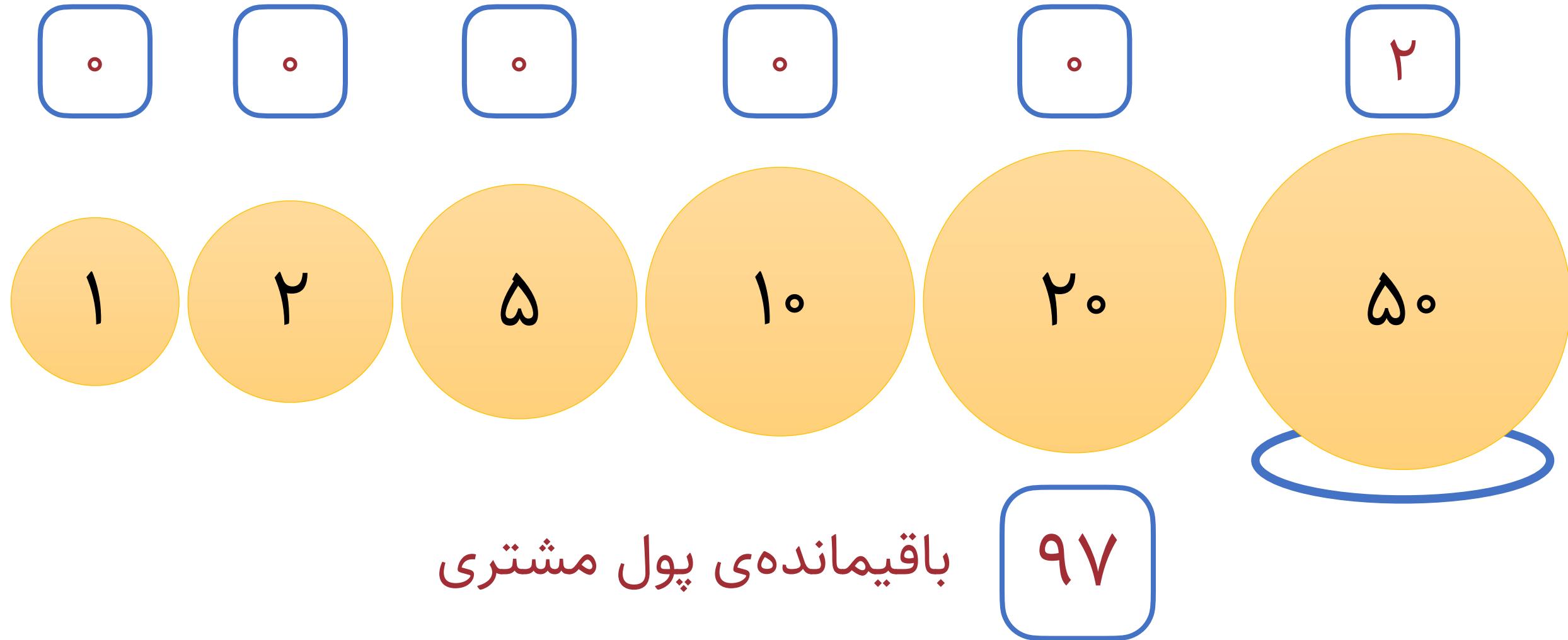
بیاید یک الگوریتم کشف کنیم!



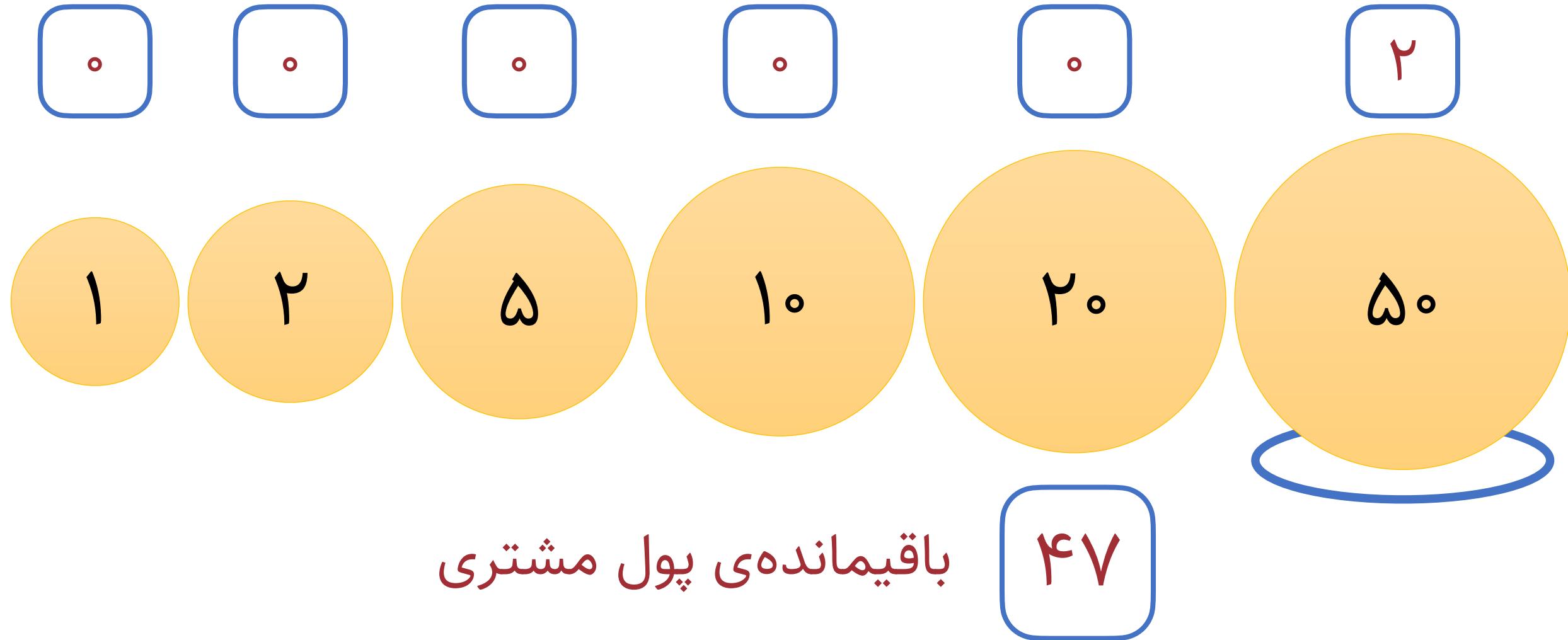
بیاید یک الگوریتم کشف کنیم!



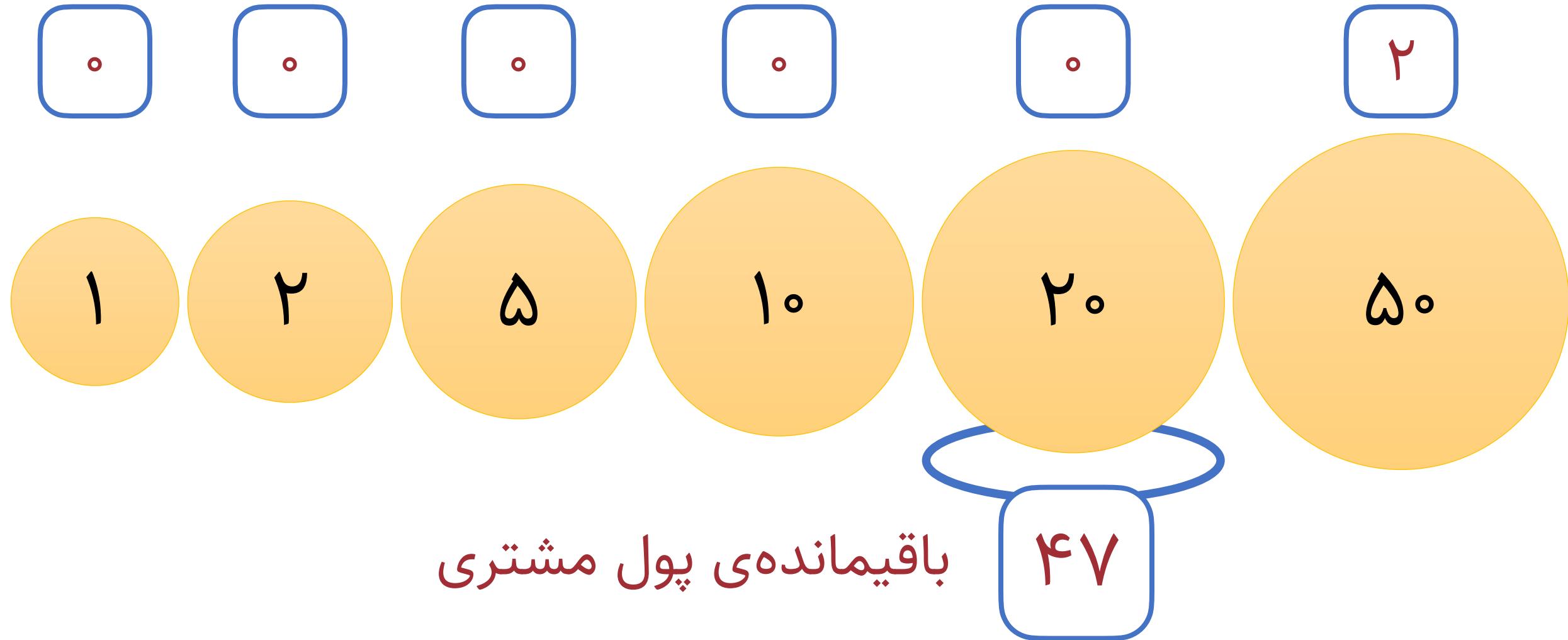
بیاید یک الگوریتم کشف کنیم!



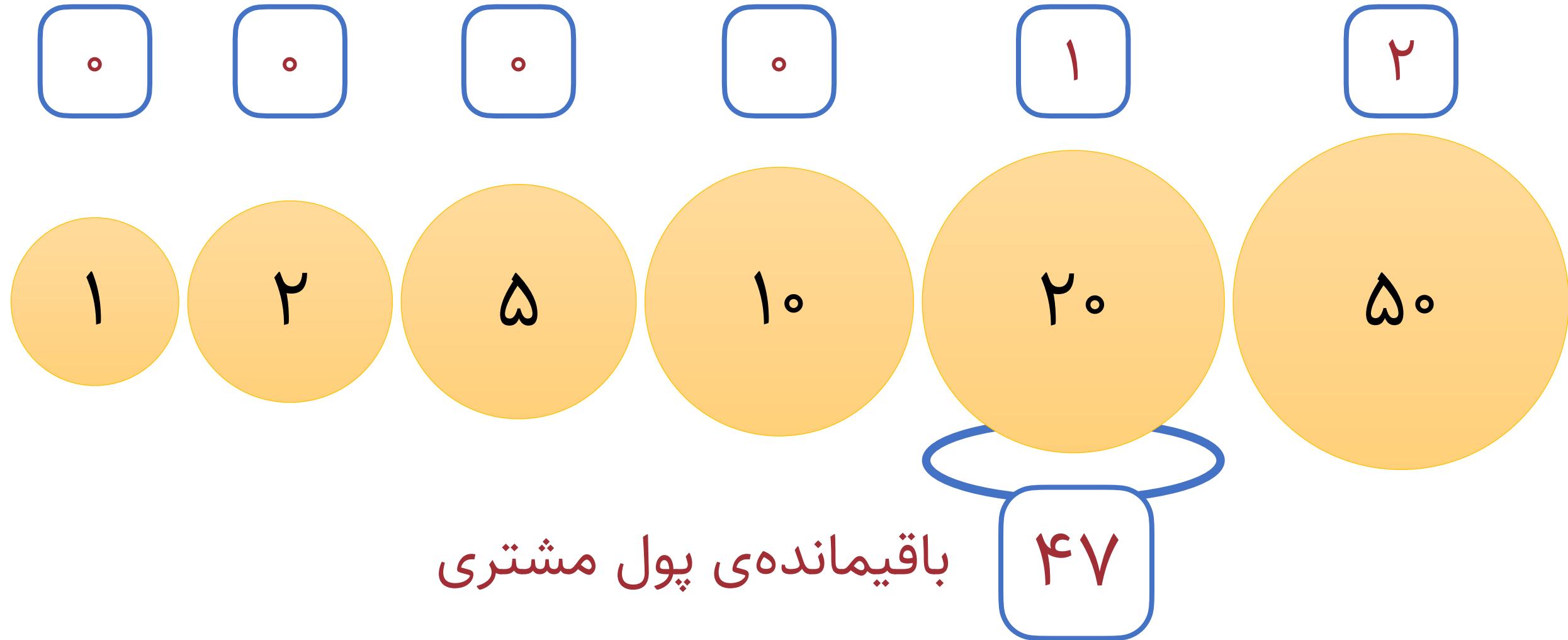
بیاید یک الگوریتم کشف کنیم!



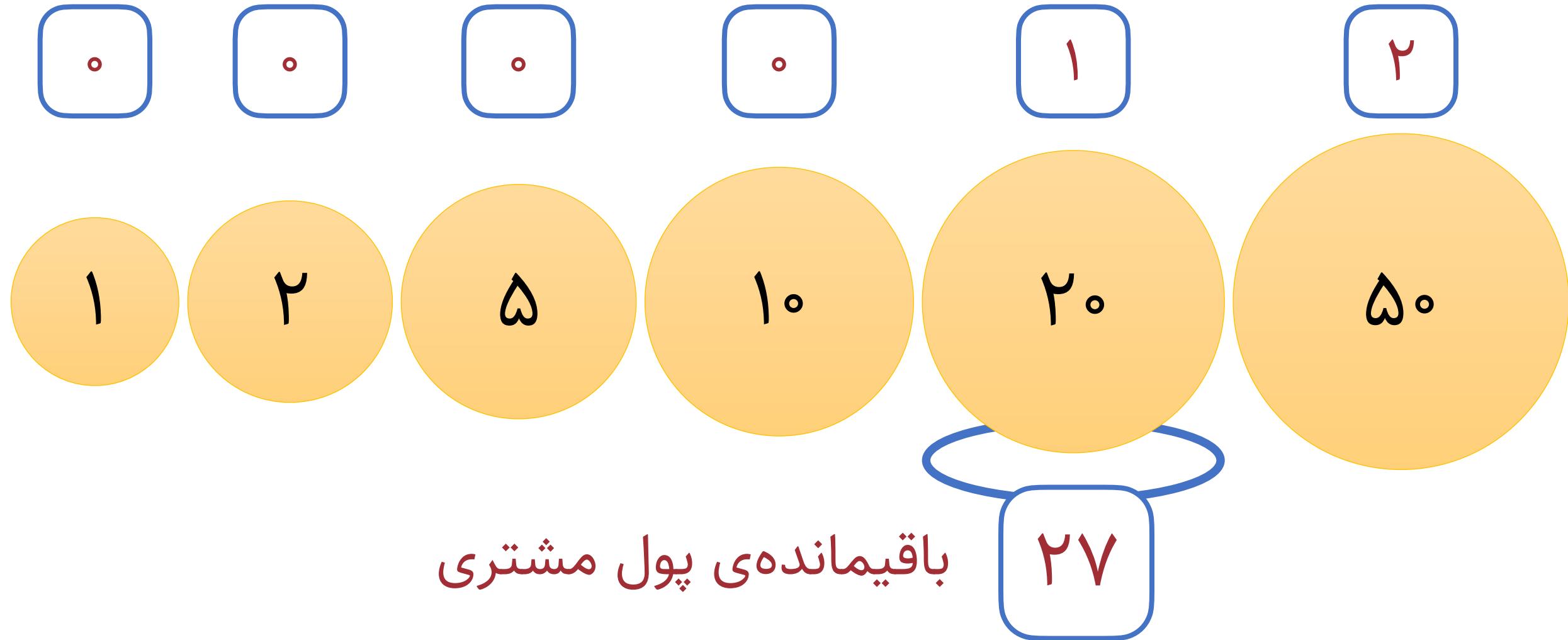
بیاید یک الگوریتم کشف کنیم!



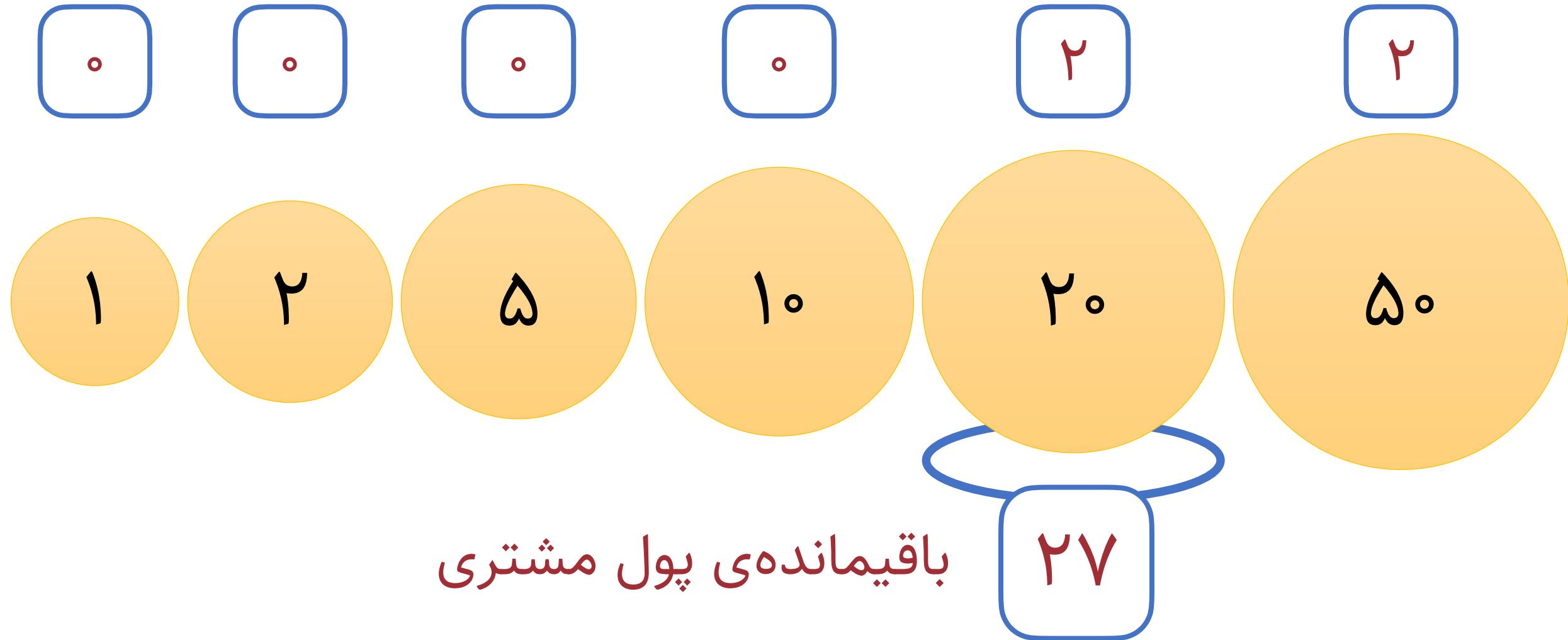
بیاید یک الگوریتم کشف کنیم!



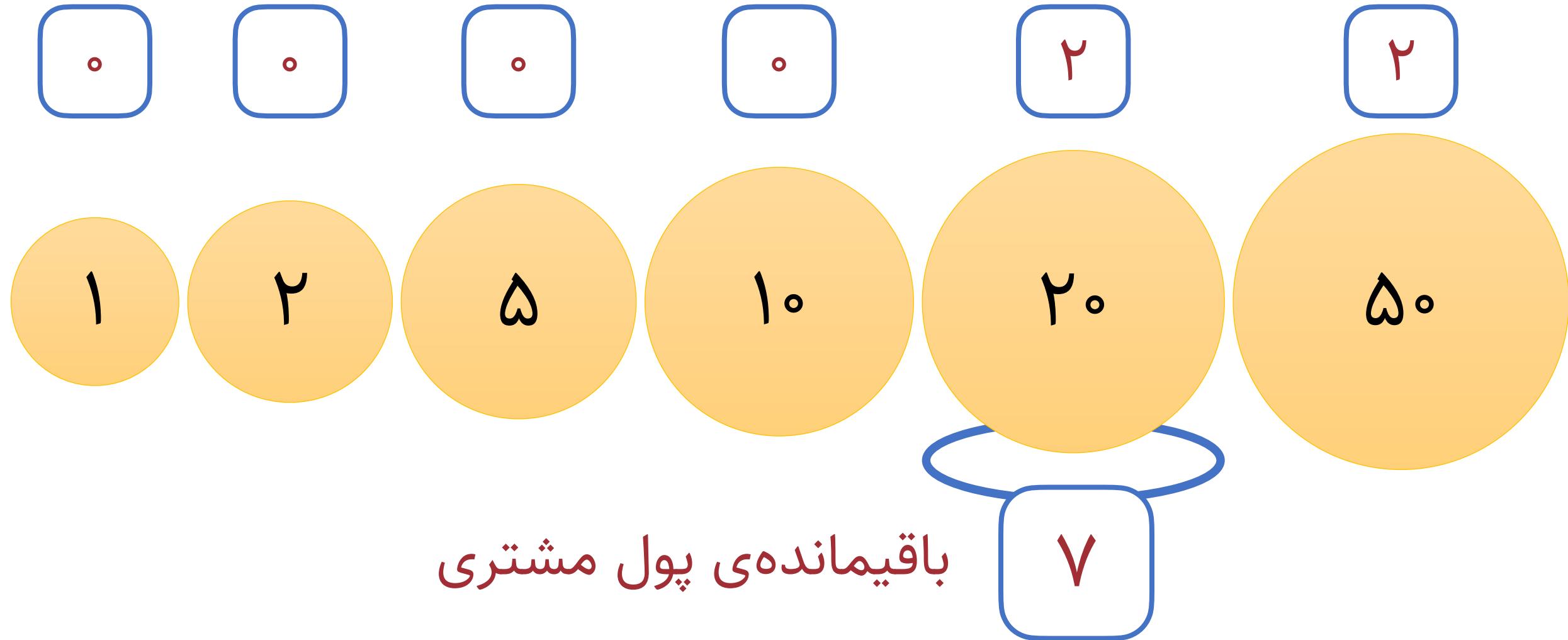
بیاید یک الگوریتم کشف کنیم!



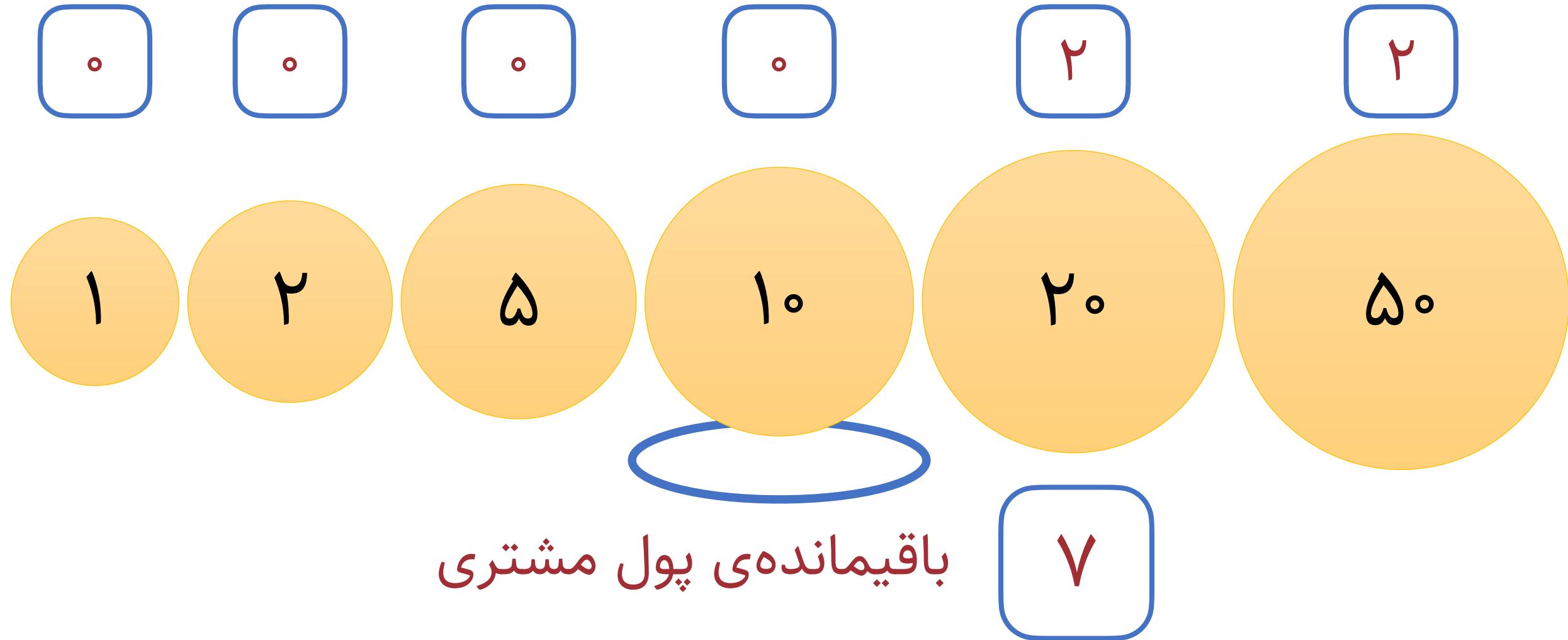
بیاید یک الگوریتم کشف کنیم!



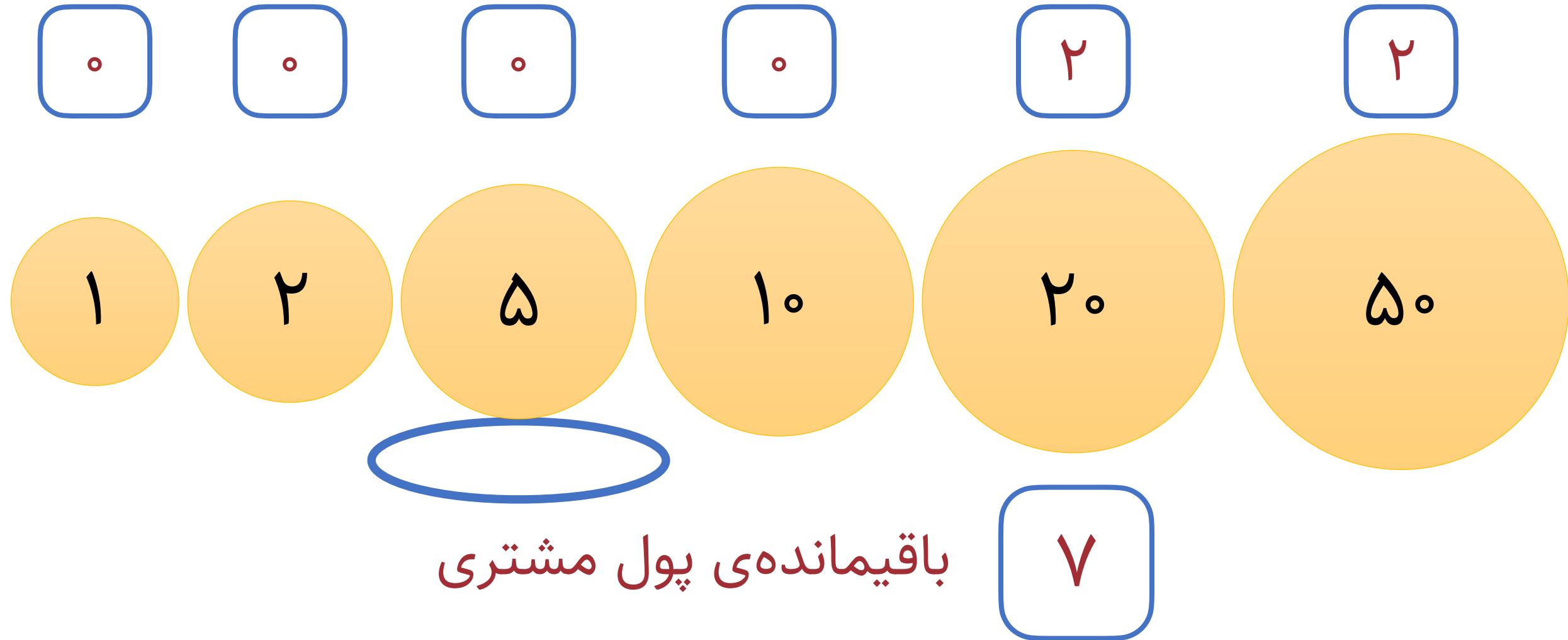
بیاید یک الگوریتم کشف کنیم!



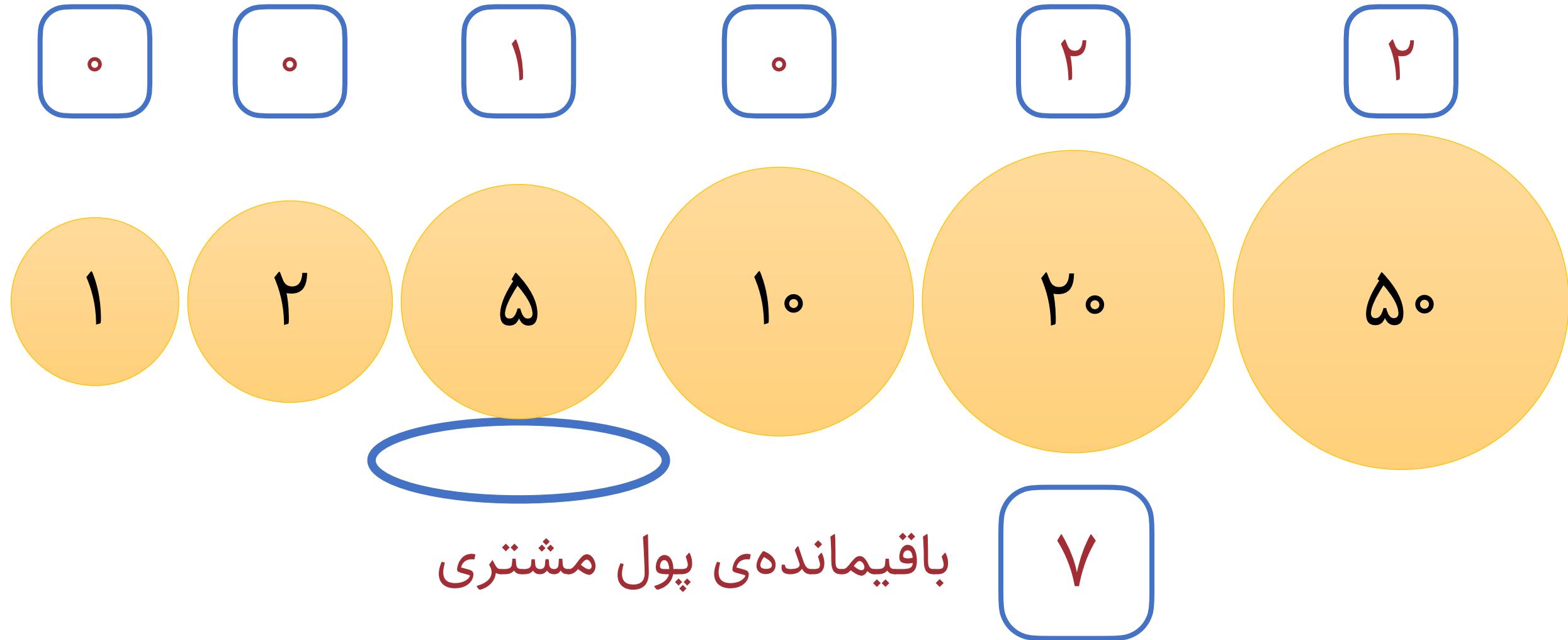
باید یک الگوریتم کشف کنیم!



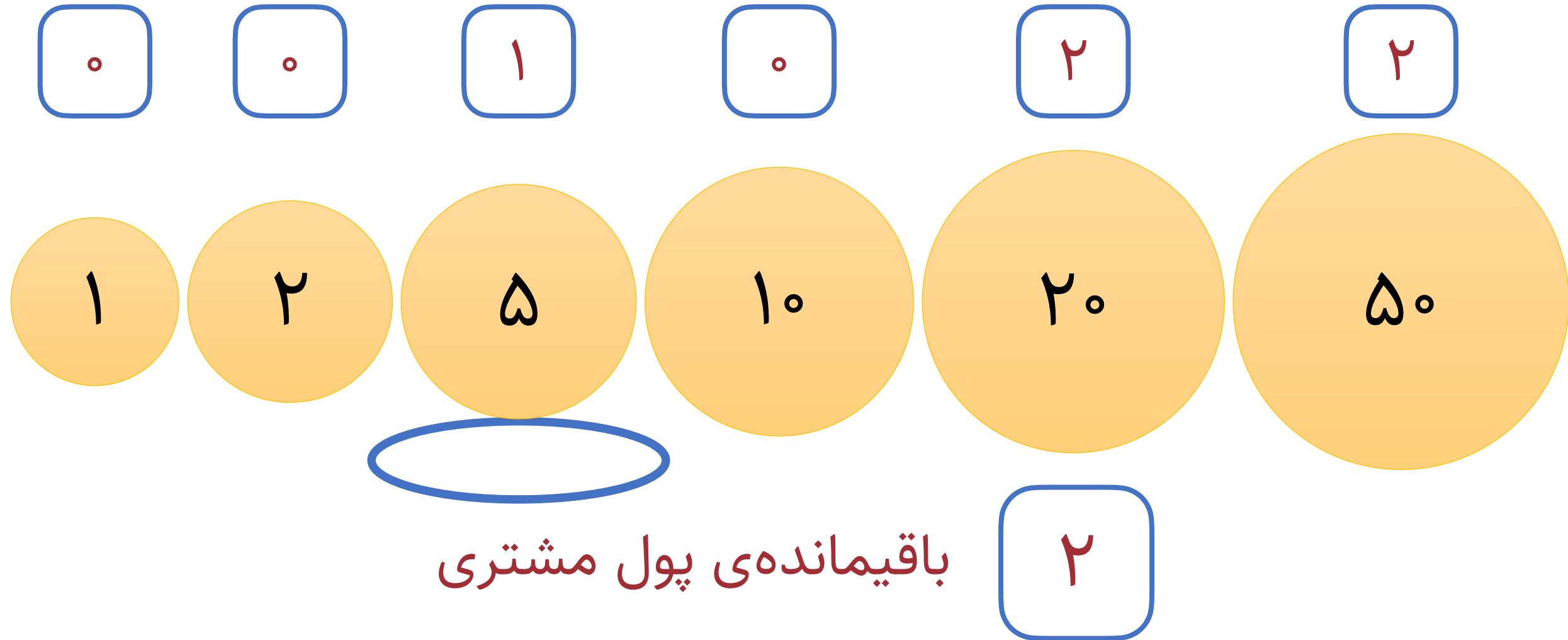
بیاید یک الگوریتم کشف کنیم!



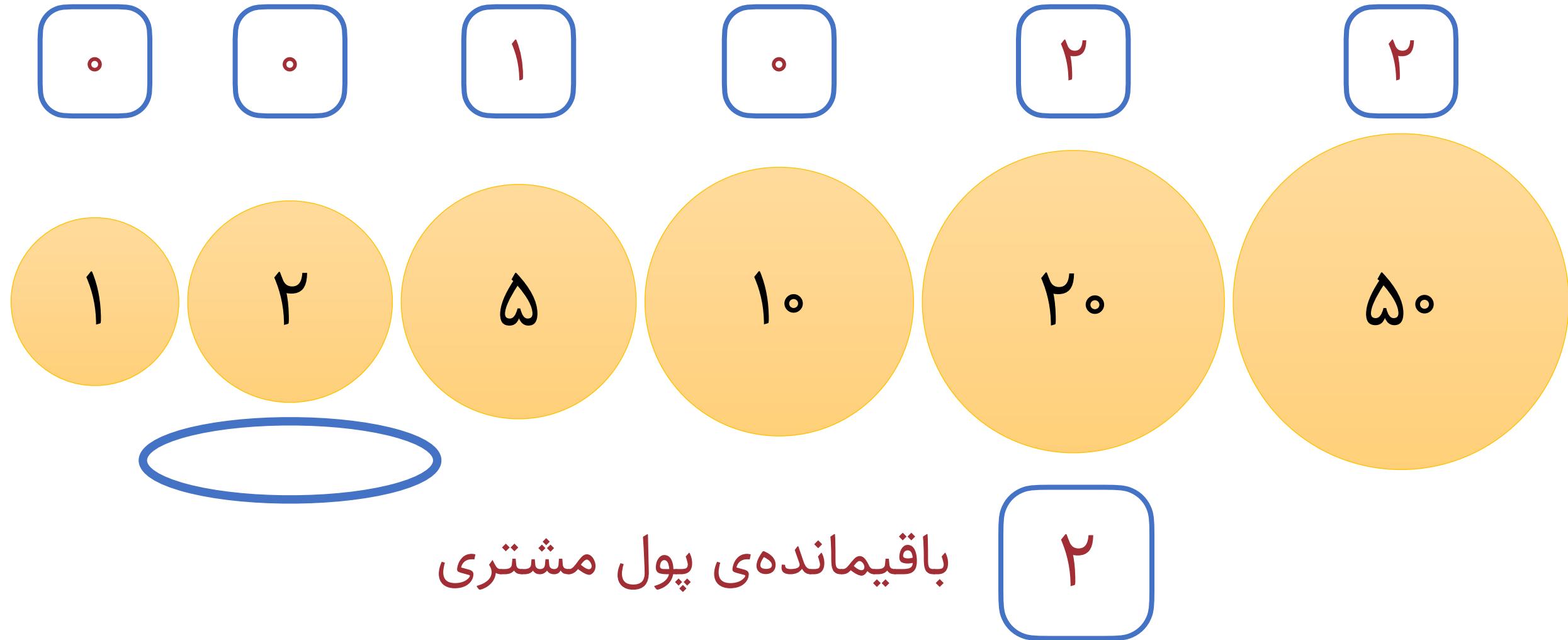
بیاید یک الگوریتم کشف کنیم!



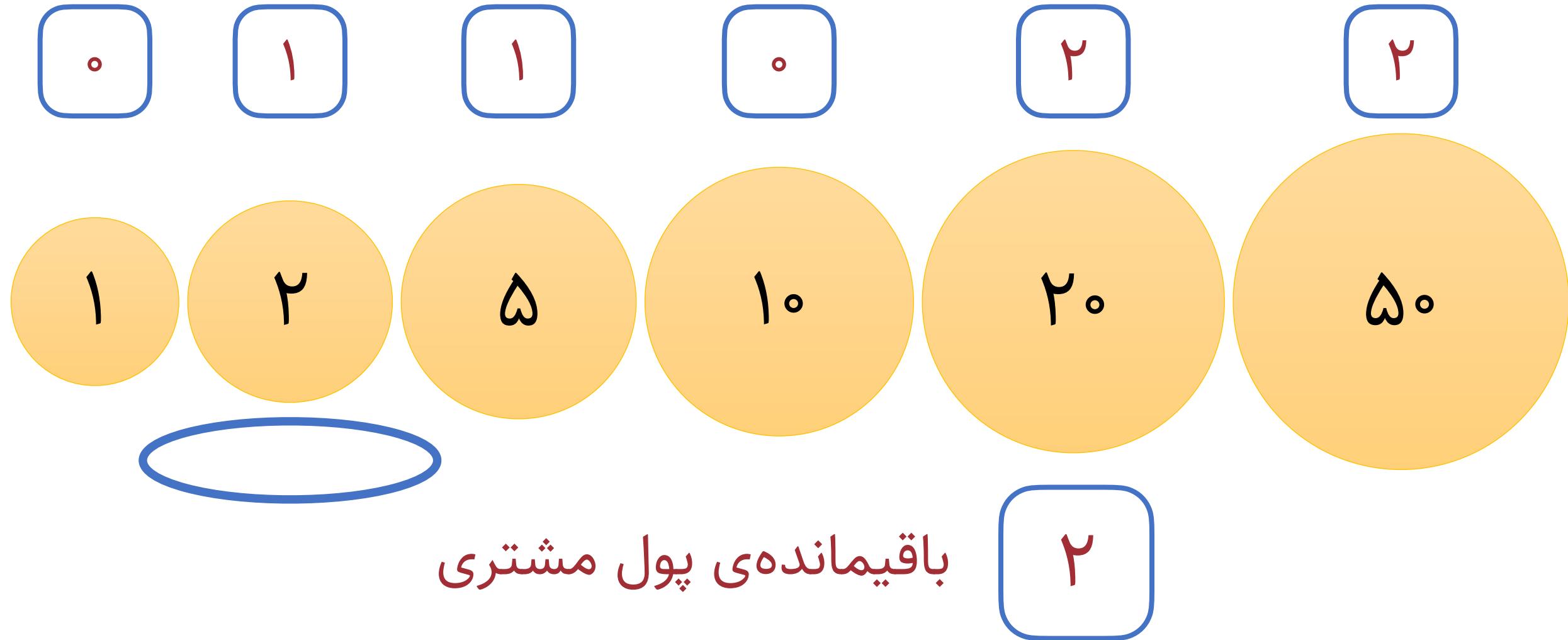
بیاید یک الگوریتم کشف کنیم!



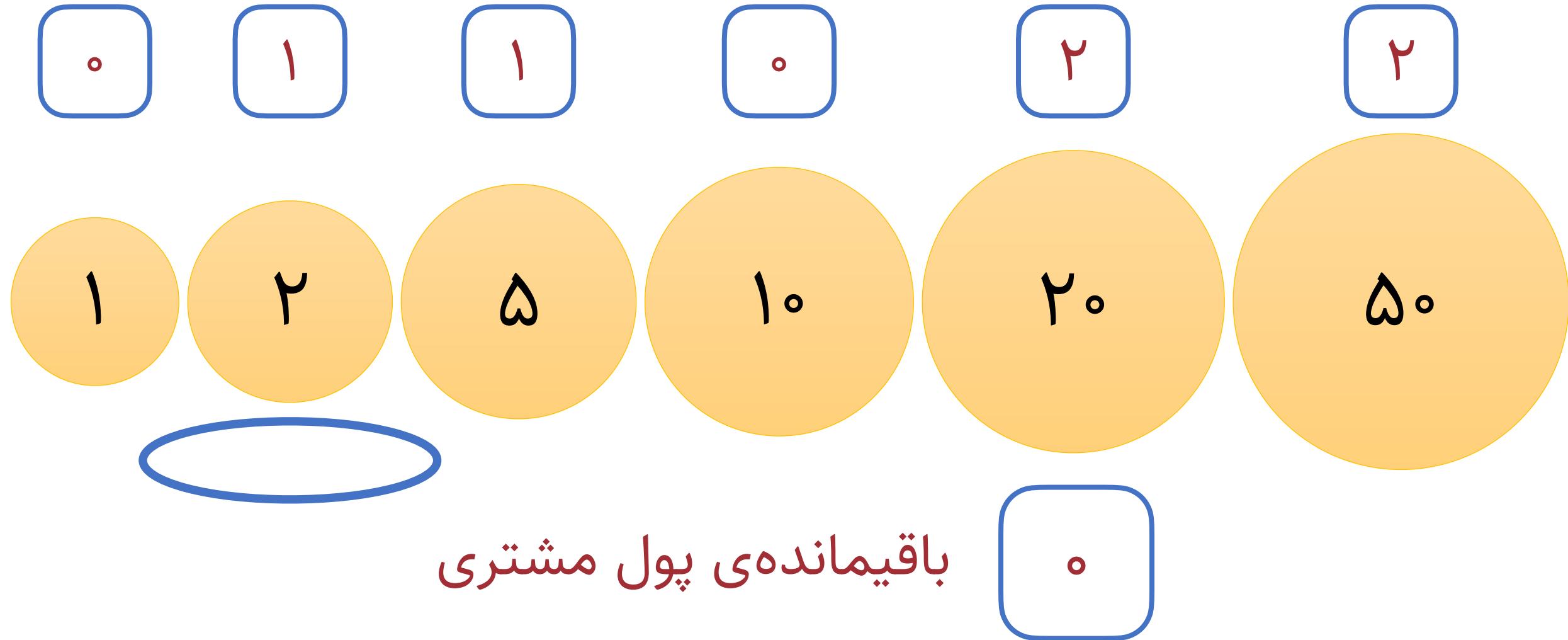
بیاید یک الگوریتم کشف کنیم!



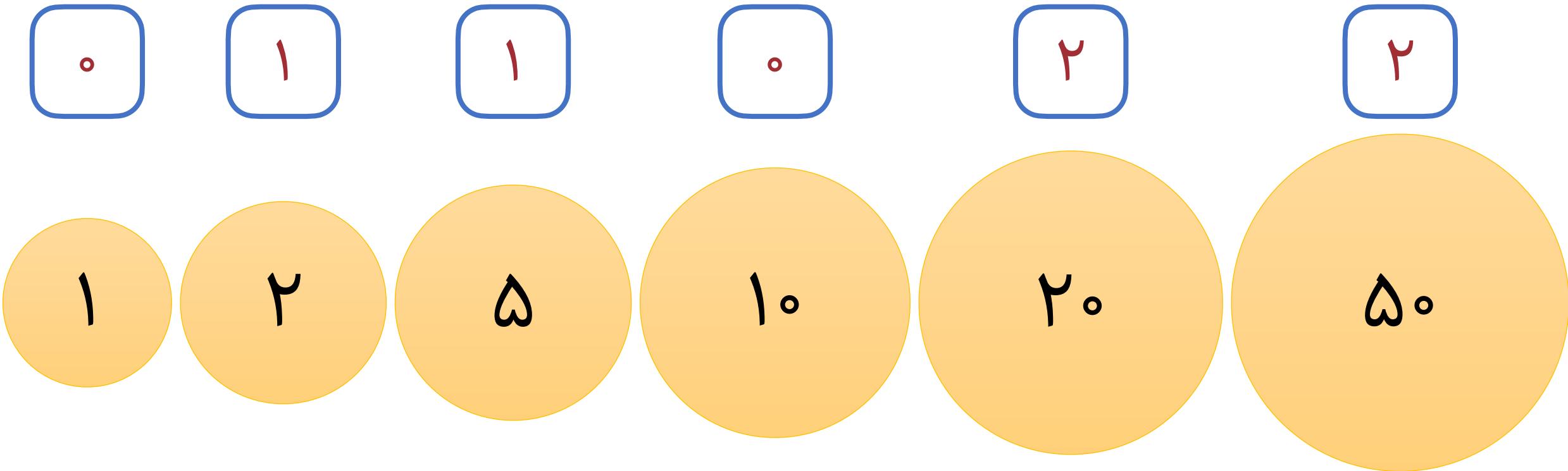
بیاید یک الگوریتم کشف کنیم!



بیاید یک الگوریتم کشف کنیم!



بیاید یک الگوریتم کشف کنیم!



باقیماندهی پول مشتری



وقت کدزدن (:)

الگوریتم حریصانه

$n[5]=0$

$n[4]=0$

$n[3]=0$

$n[2]=0$

$n[1]=0$

$n[0]=0$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$

$m = 147$

```
for i in range(6):
```

n[5]=0

n[4]=0

n[3]=0

n[2]=0

n[1]=0

n[0]=0

c[5]=1

c[4]=2

c[3]=5

c[2]=10

c[1]=20

c[0]=50

m = 147

i=0

while $m > c[i]$:

$n[5]=0$

$n[4]=0$

$n[3]=0$

$n[2]=0$

$n[1]=0$

$n[0]=0$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$

$m = 147$

$i=0$

الگوریتم حریصانه

$$n[i] = n[i]+1$$

n[5]=0

n[4]=0

n[3]=0

n[2]=0

n[1]=0

n[0]=1

c[5]=1

c[4]=2

c[3]=5

c[2]=10

c[1]=20

c[0]=50

m = 147

i=0

الگوریتم حریصانه

$$m = m - c[i]$$

$n[5]=0$

$n[4]=0$

$n[3]=0$

$n[2]=0$

$n[1]=0$

$n[0]=1$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$

$$m = 97$$

$i=0$

while $m > c[i]$:

$n[5]=0$

$n[4]=0$

$n[3]=0$

$n[2]=0$

$n[1]=0$

$n[0]=1$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$

$m = 97$

$i=0$

الگوریتم حریصانه

$$n[i] = n[i]+1$$

n[5]=0

n[4]=0

n[3]=0

n[2]=0

n[1]=0

n[0]=2

c[5]=1

c[4]=2

c[3]=5

c[2]=10

c[1]=20

c[0]=50

m = 97

i=0

الگوریتم حریصانه

$$m = m - c[i]$$

$n[5]=0$

$n[4]=0$

$n[3]=0$

$n[2]=0$

$n[1]=0$

$n[0]=2$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$

$$m = 47$$

$i=0$

while $m > c[i]$:

$n[5]=0$

$n[4]=0$

$n[3]=0$

$n[2]=0$

$n[1]=0$

$n[0]=2$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$

$m =$

47

$i=0$

```
for i in range(6):
```

n[5]=0

n[4]=0

n[3]=0

n[2]=0

n[1]=0

n[0]=2

c[5]=1

c[4]=2

c[3]=5

c[2]=10

c[1]=20

c[0]=50

i=1

m = 47

while $m > c[i]$:

$n[5]=0$

$n[4]=0$

$n[3]=0$

$n[2]=0$

$n[1]=0$

$n[0]=2$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$

$m = 47$

$i=1$

الگوریتم حریصانه

$$n[i] = n[i]+1$$

n[5]=0

n[4]=0

n[3]=0

n[2]=0

n[1]=1

n[0]=2

c[5]=1

c[4]=2

c[3]=5

c[2]=10

c[1]=20

c[0]=50

m = 47

i=1

$$m = m - c[i]$$

n[5]=0

n[4]=0

n[3]=0

n[2]=0

n[1]=1

n[0]=2

c[5]=1

c[4]=2

c[3]=5

c[2]=10

c[1]=20

c[0]=50

$$m = 27$$

i=1

while $m > c[i]$:

$n[5]=0$

$n[4]=0$

$n[3]=0$

$n[2]=0$

$n[1]=1$

$n[0]=2$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$

$m =$

27

i=1

الگوریتم حریصانه

$n[i] = n[i]+1$

$n[5]=0$

$n[4]=0$

$n[3]=0$

$n[2]=0$

$n[1]=2$

$n[0]=2$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$

$m = 27$

$i=1$

$$m = m - c[i]$$

n[5]=0

n[4]=0

n[3]=0

n[2]=0

n[1]=2

n[0]=2

c[5]=1

c[4]=2

c[3]=5

c[2]=10

c[1]=20

c[0]=50

m =

7

i=1

while $m > c[i]$:

$n[5]=0$

$n[4]=0$

$n[3]=0$

$n[2]=0$

$n[1]=2$

$n[0]=2$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$

$m =$

7

$i=1$

for i in range(6):

n[5]=0

n[4]=0

n[3]=0

n[2]=0

n[1]=2

n[0]=2

c[5]=1

c[4]=2

c[3]=5

c[2]=10

c[1]=20

c[0]=50

i=2

m =

7

while $m > c[i]$:

$n[5]=0$

$n[4]=0$

$n[3]=0$

$n[2]=0$

$n[1]=2$

$n[0]=2$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$

$i=2$

$m =$

7

for i in range(6):

n[5]=0

n[4]=0

n[3]=0

n[2]=0

n[1]=2

n[0]=2

c[5]=1

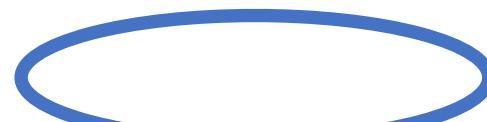
c[4]=2

c[3]=5

c[2]=10

c[1]=20

c[0]=50



i=3

m =

7

while $m > c[i]$:

$n[5]=0$

$n[4]=0$

$n[3]=0$

$n[2]=0$

$n[1]=2$

$n[0]=2$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$

$m =$

7

$i=3$

$$n[i] = n[i]+1$$

n[5]=0

n[4]=0

n[3]=1

n[2]=0

n[1]=2

n[0]=2

c[5]=1

c[4]=2

c[3]=5

c[2]=10

c[1]=20

c[0]=50



i=3

m =

7

$$m = m - c[i]$$

$n[5]=0$

$n[4]=0$

$n[3]=1$

$n[2]=0$

$n[1]=2$

$n[0]=2$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$



$i=3$

$m =$

2

while $m > c[i]$:

$n[5]=0$

$n[4]=0$

$n[3]=1$

$n[2]=0$

$n[1]=2$

$n[0]=2$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$



$i=3$

$m =$

2

```
for i in range(6):
```

n[5]=0

n[4]=0

n[3]=1

n[2]=0

n[1]=2

n[0]=2

c[5]=1

c[4]=2

c[3]=5

c[2]=10

c[1]=20

c[0]=50



i=4

m = 2

while $m > c[i]$:

$n[5]=0$

$n[4]=0$

$n[3]=1$

$n[2]=0$

$n[1]=2$

$n[0]=2$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$



$i=4$

$m = 2$

$$n[i] = n[i]+1$$

n[5]=0

n[4]=1

n[3]=1

n[2]=0

n[1]=2

n[0]=2

c[5]=1

c[4]=2

c[3]=5

c[2]=10

c[1]=20

c[0]=50



i=4

m = 2

$$m = m - c[i]$$

$n[5]=0$

$n[4]=1$

$n[3]=1$

$n[2]=0$

$n[1]=2$

$n[0]=2$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$



$i=4$

$m = 0$

while $m > c[i]$:

$n[5]=0$

$n[4]=1$

$n[3]=1$

$n[2]=0$

$n[1]=2$

$n[0]=2$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$



$i=4$

$m = 0$

```
for i in range(6):
```

n[5]=0

n[4]=1

n[3]=1

n[2]=0

n[1]=2

n[0]=2

c[5]=1

c[4]=2

c[3]=5

c[2]=10

c[1]=20

c[0]=50



i=5

m = 0

while $m > c[i]$:

$n[5]=0$

$n[4]=1$

$n[3]=1$

$n[2]=0$

$n[1]=2$

$n[0]=2$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$



$i=5$

$m = 0$

```
for i in range(6):
```

n[5]=0

n[4]=1

n[3]=1

n[2]=0

n[1]=2

n[0]=2

c[5]=1

c[4]=2

c[3]=5

c[2]=10

c[1]=20

c[0]=50

m = 0

الگوریتم حریصانه

$n[5]=0$

$n[4]=1$

$n[3]=1$

$n[2]=0$

$n[1]=2$

$n[0]=2$

$c[5]=1$

$c[4]=2$

$c[3]=5$

$c[2]=10$

$c[1]=20$

$c[0]=50$

$m = 0$

وقت کدزدن (:)

زبل می خواهد کیک بخرد.



هر کیک قیمتی دارد. زبل هم در جیبش مقداری پول دارد.



۱۸

۲۳

۵۳

۳۸

۴۰

۲۵

۱۳

پول زبل

۹۵

زبل می خواهد با پولش بیشترین تعداد کیک را بخرد.



۱۸

۲۳

۵۳

۳۸

۴۰

۲۵

۱۳

پول زبل

۹۵

ابتدا کیک‌ها را به ترتیب قیمت مرتب می‌کند.



۱۳

۱۸

۲۳

۲۵

۳۸

۴۰

۵۳

پول زبل

۹۵

سپس به صورت حریصانه، کیک با کمترین قیمت را می خرد.



۱۳



۱۸



۲۳



۲۵



۳۸



۴۰



۵۳

پول زبل

۹۵

سپس به صورت حریصانه، کیک با کمترین قیمت را می خرد.



۱۳



۱۸



۲۳



۲۵



۳۸



۴۰



۵۳

باقی مانده‌ی پول زبل

۸۲

و همین کار را تکرار می‌کند...



۱۳



۱۸



۲۳



۲۵



۳۸



۴۰



۵۳

باقی‌مانده‌ی پول زبل

۸۲

و همین کار را تکرار می‌کند...



باقی‌ماندهی پول زبل

۸۲

و همین کار را تکرار می‌کند...



باقی‌ماندهی پول زبل

۶۴

و همین کار را تکرار می‌کند...



باقی‌مانده‌ی پول زبل

۶۴

و همین کار را تکرار می‌کند...



باقی‌ماندهی پول زبل

۴۱

و همین کار را تکرار می‌کند...



باقی‌ماندهی پول زبل

۴۱

و همین کار را تکرار می‌کند...



باقی‌مانده‌ی پول زبل

۱۶

تا همهی کیک‌ها را بخرد یا باقی‌ماندهی پولش به اندازه‌ی قیمت کیک بعدی نباشد.



۱۳



۱۸



۲۳



۲۵



۳۸



۴۰



۵۳

باقی‌ماندهی پول زبل

۱۶

چرا الگوریتم زبل، بیشترین تعداد کیک ممکن برای خرید را می‌یابد؟



باقی‌مانده‌ی پول زبل

۱۶

فرض کنید من جواب دیگری پیدا کنم.



۱۳



۱۸



۲۳



۲۵



۳۸



۴۰



۵۳

باقی مانده‌ی پول زبل

۳

اگر یکی از کیکهایی که الگوریتم زبل انتخاب کرد در جواب من نباشد...



باقی مانده‌ی پول زبل

۳

و تعداد کیکهایی که من خریده ام بیشتر یا مساوی کیکهای زبل باشد.



باقی‌مانده‌ی پول زبل

۳

در آن صورت زبل می‌تواند به جای یکی از کیک‌های من کیک خودش را بخرد...



باقی‌مانده‌ی پول زبل

۳

در آن صورت زبل می‌تواند به جای یکی از کیک‌های من کیک خودش را بخرد...



۱۳



۱۸



۲۳



۲۵



۳۸



۴۰



۵۳

باقی‌مانده‌ی پول زبل

۱۶

طوری که هزینه‌ی بیشتری هم پرداخت نکند.



باقی‌مانده‌ی پول زبل

۱۶

پس تعداد کیکهایی که من خریده ام نمیتواند بیش از کیکهای زبل باشد.



باقی مانده‌ی پول زبل

۱۶

وقت کدزدن (:)

شاد و تندرنست باشید :-)