

## سرعت رشد

- توابع پیچیدگی زیر را از لحاظ سرعت رشد مرتب کنید.

$$2^{\log n}$$

$$e^n$$

$$\log n^2$$

$$100n^n + \pi$$

$$e^{20}$$

$$10n\sqrt{n}$$

$$n^{15} + 20n + 25\log n$$

$$n! + n$$

$$(\log n)^2$$

$$12(\log n)!$$

$$\frac{1}{5}n^{\left(\frac{1}{\log n}\right)}$$

$$\ln(\ln n)$$

$$\sqrt{\log n}$$

## صحت روابط

- صحت روابط زیر را اثبات کنید.
- توجه: توابع پیچیدگی را صعودی در نظر بگیرید.

$$(n + \log n)^3 \in \theta(n^3)$$

$$n\sqrt{n} \in O(n^n)$$

$$\sum_{i=1}^n \sum_{j=1}^i j \in \theta(n^3)$$

$$f(n) + o(f(n)) \in \theta(f(n))$$

$$f(n) + g(n) \in O(\max(f(n), g(n)))$$

$$f^2(n) \in \Omega(f(n))$$

## درستی یا نادرستی

- با فرض مثبت بودن توابع  $f$  و  $g$ ، درستی یا نادرستی روابط زیر را مشخص کنید.
- برای درستی بیان اثبات و برای نادرستی تنها ارائه‌ی یک مثال نقض کافی است.
- **راهنمایی:** برای در نظر گرفتن توابع، همه‌ی توابع را در نظر بگیرید؛ نه صرفاً توابع صعودی.

$$f(n) \in O(g(n)) \rightarrow 2^{f(n)} \in O(2^{g(n)})$$

$$f(n) \in O(f(n)^2)$$

$$f(n) + g(n) \in \theta(\min(f(n), g(n)))$$

$$f(n) \in \theta(f(\frac{n}{2}))$$

$$f(n) \in O(g(n)) \rightarrow \log(f(n)) \in O(\log(g(n)))$$

where  $\log(g(n)) \geq 0$ ,  $f(n) > 0$  and  $f(n) \geq 1$  for all sufficiently large  $n$

## مقایسه

- الگوریتم‌های  $A$  و  $B$  دقیقاً  $T_A(n)$  و  $T_B(n)$  میکرو ثانیه برای حل مسئله‌ای با اندازه‌ی  $n$  طول می‌کشند.

$$T_A(n) = 0.1n^2 \log n$$

$$T_B(n) = 2.5n^2$$

- از بین این دو الگوریتم، الگوریتمی را انتخاب کنید که از نظر  $O$  بهتر باشد.
- کوچکترین  $n_0$  ممکن را پیدا کنید که به ازای هر  $n \geq n_0$  یک الگوریتم از نظر زمان اجرا بهتر از دیگری باشد.
- اگر  $n \leq 512$  باشد، کدام الگوریتم به صرفه‌تر است؟

## پیچیدگی کدها

- پیچیدگی زمانی تکه کدهای زیر را بدست آورید.

```
1 for i in range(n):
2     j = 0
3     while j <= n:
4         print(i, j)
5         j += n // 50
```

```
1 i = n
2 while i >= 1:
3     j = i
4     while j <= n:
5         print(i)
6         j *= 2
7     i //= 2
```

```
1 i = 2
2 while i <= n:
3     print("*")
4     i = i * i
```

```
1 for i in range(1, n+1):
2     j = 1
3     while j <= i:
4         print("*")
5         j *= 2
```

- توجه: علامت "//"، علامت تقسیم با خارج قسمت صحیح است.

## حداقل ضرب

- با **حداقل** چه تعداد ضرب می‌توان  $x^{64}$  را محاسبه کرد؟
- ایده‌ی حل را به صورت کامل توضیح دهید.

## جایگشت خوب یا سوال خوب؟

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۵۰ مگابایت
- یک جایگشت به طول  $n$  دنباله‌ای از اعداد مجزای 1 تا  $n$  با ترتیبی دلخواه است. به عنوان مثال  $[1, 3, 2, 5, 4]$  یک جایگشت است اما  $[1, 2, 2]$  یک جایگشت نیست (عدد 2 دوبار تکرار شده است) و  $[1, 3, 4]$  هم یک جایگشت نیست (عدد 4 ظاهر شده در صورتی که  $n$  برابر با 3 است).
- به ازای عدد طبیعی مثبت  $n$  جایگشت  $p$  را یک جایگشت خوب میگوییم اگر به ازای هر  $i$  و  $j$  (که  $j \leq i$ ) داشته باشیم (که منظور از  $OR$ ، bitwise OR operation است):

$$(p_i \text{ OR } p_{i+1} \text{ OR } \dots \text{ OR } p_{j-1} \text{ OR } p_j) \geq j - i + 1$$

- به عبارت دیگر یک جایگشت خوب است اگر و تنها اگر به ازای هر زیر دنباله از آن،  $OR$  همه‌ی اعداد آن زیر دنباله از تعداد اعداد موجود در آن کمتر نباشد.
- حال، برنامه‌ای بنویسید که با گرفتن عدد طبیعی  $n$  یکی از جایگشت‌های خوب به طول  $n$  را در خروجی چاپ کند.

## ورودی

- ورودی تنها شامل یک خط است که در آن عدد طبیعی  $n$  آمده است.

$$1 \leq n \leq 100$$

## خروجی

- خروجی برنامه‌ی شما باید شامل یک خط باشد که در آن یک جایگشت خوب به طول  $n$  آمده است.

## ورودی نمونه



5

خروجی نمونه

1 3 2 5 4

## واحد حوصله سربر -

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۵۰ مگابایت
- یک ساختمان گول آسا (!) مفروض است با 10000 واحد آپارتمان، و هر کدام از واحدها شناسه‌ای بین 1 تا 10000 دارند.
- تعریف می‌کنیم واحدی حوصله سربر است که همه‌ی رقم‌های آن یکی باشند! به طور مثال واحدهای 55، 2، 9999 و 111 حوصله سربر هستند.
- آقای صورتی وظیفه دارد به تمامی واحدهای حوصله سربر تلفن بزند تا زمانی که یکی از آن‌ها جواب دهد! و این کار را به ترتیب زیر انجام می‌دهد:
  - ابتدا به تمام واحدهای حوصله سربر متشکل از رقم 1، به صورت صعودی زنگ می‌زند: اول 1، سپس 11، سپس 111 و در نهایت 1111.
  - سپس به تمام واحدهای حوصله سربر متشکل از رقم 2، به صورت صعودی زنگ می‌زند: اول 2، سپس 22، سپس 222 و در نهایت 2222.
  - و به همین ترتیب ادامه می‌دهد.
  - در این بین اگر واحد  $x$  جواب تلفن را بدهد آقای صورتی بس می‌کند و دیگر به واحدهای بعدی زنگ نمی‌زند!
- می‌خواهیم حساب کنیم که آقای صورتی جمعاً چند رقم را فشرده است و از شما می‌خواهیم این مقدار را حساب کنید. به طور مثال اگر واحد 222 جواب بدهد آنگاه خواهیم داشت:

$$1 + 2 + 3 + 4 + 1 + 2 + 3 = 16$$

(چرا که به ترتیب از چپ به راست با واحد های روبرو تماس گرفته است:  
 (1, 11, 111, 1111, 2, 22, 222

**ورودی**

- خط اول ورودی شامل  $t$  (تعداد تست کیس‌ها) است.

$$1 \leq t \leq 36$$

- در خطوط بعدی در هر خط عدد  $x$  آمده است که نشان دهنده‌ی شماره آپارتمانی است که پاسخ داده است. همچنین تضمین می‌شود  $x$  ارقام یکسانی دارد.

$$1 \leq x \leq 9999$$

## خروجی

- به ازای هر تست کیس، تعداد ارقامی که آقای صورتی فشرده است را در یک خط چاپ کنید.

## ورودی نمونه

4  
22  
9999  
1  
777

## خروجی نمونه

13  
90  
1  
66

## مجموع بازه

- محدودیت زمان: ۵ ثانیه
- محدودیت حافظه: ۵۰ مگابایت
- یک آرایه با طول  $n$  را در نظر بگیرید، فرض کنید  $Q$  درخواست شامل زوج‌های  $(i, j)$  داریم.
- برنامه‌ای بنویسید که به ازای هر درخواست جمع اعداد آرایه بین اندیس‌های  $i$  و  $j$  را محاسبه کند (اندیس‌ها از صفر شروع می‌شوند).

## ورودی

- در خط اول ورودی، عدد طبیعی  $n$  آمده است.

$$1 \leq n \leq 50000$$

- در خط دوم ورودی،  $n$  عدد حسابی که مقادیر آرایه هستند، آمده است.

$$0 \leq a[i] \leq 100000$$

- در خط سوم ورودی، عدد طبیعی  $Q$  که تعداد درخواست هاست، آمده است.

$$1 \leq Q \leq 500000$$

- در  $Q$  خط بعدی در هر خط، دو عدد حسابی  $i$  و  $j$  آمده است.

$$0 \leq i, j < n$$

## خروجی

- در خروجی  $Q$  خط چاپ کنید که در هر خط جمع مقادیر آرایه بین اندیس‌های  $i$  و  $j$  آمده باشد.

## ورودی نمونه

5  
1 3 2 7 1  
3  
4 2  
1 1  
1 3

## خروجی نمونه

10  
3  
12

## عدد گمشده

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲ مگابایت
- آرایه‌ی  $a$  را در نظر بگیرید، در این آرایه  $n$  عدد صحیح متمایز از بازه‌ی  $[0, n]$  وجود دارد.
- برنامه‌ای بنویسید که پیدا کند کدام یک از اعداد بازه‌ی  $[0, n]$  در آرایه‌ی  $a$  نیست.
- توجه: به محدودیت حافظه‌ی غیر معمول در این سوال دقت کنید!

## ورودی

- در خط اول ورودی، عدد طبیعی  $n$  آمده است.

$$1 \leq n \leq 1000000$$

- در خط دوم ورودی،  $n$  عدد حسابی که مقادیر آرایه هستند، آمده است.

$$0 \leq a[i] \leq n$$

## خروجی

- در تنها خط خروجی عدد خواسته شده در صورت سوال را چاپ کنید.

## ورودی نمونه

5  
4 2 3 5 0

## خروجی نمونه

1