

به نام خدا

امتحان دوم درس طراحی و تحلیل الگوریتم‌ها (بهار ۱۴۰۰)

مدت امتحان: ۹۰ دقیقه

امتحان شامل سه بخش است و از هر بخش شما باید فقط و فقط یک سوال را حل کنید. برای انتخاب سوال مربوط به خود به توضیحات بالای هر بخش توجه کنید. برای انتخاب سوال خود فرض کنید شماره دانشجویی شما $ABCDEFGHI$ باشد و مقادیر A تا I را با توجه به شماره دانشجویی خود در هر بخش جایگزین کنید. برای مثال اگر شماره دانشجویی شما 894456123 است مقدار $A = 8$ و مقدار $H = 2$ خواهد بود. لطفا در بالای برگه جواب حتما علاوه بر نام و نام خانوادگی شماره دانشجویی و شماره سوال مربوط به خود را بنویسید

بخش اول (۲۵ نمره): با توجه به شماره دانشجویی خود سوال زیر را حل کنید

۱. هدف از این سوال اجرای حل مسئله کوله‌پشتی برای 6 جنس و یک کوله‌پشتی به حجم 15 است. ارزش و حجم اجناس به شرح زیر است. مقادیر A تا I را با توجه به شماره دانشجویی خود جایگزین کنید:

اندیس (W)	6	5	4	3	2	1
ارزش (W)	2	5	4	3	5	7
حجم (V)	B	H	I	$G + I$	$H + G$	$I + G$

با توجه به ورودی بالا، دو پیاده‌سازی زیر برای مسئله کوله‌پشتی را در نظر بگیرید و به سوالات زیر جواب دهید:

(آ) اگر تنها به ورودی سوال توجه کنید کدام الگوریتم سریع‌تر خواهد بود؟ به طور مشخص تعداد اجرای خط ۶ الگوریتم ۱ را با تعداد اجرای خط ۷ الگوریتم ۲ مقایسه کنید.

(ب) بعد از اجرای $alg2(6, 15)$ مقدار آرایه‌های $B[6, 12]$ ، $B[5, 10]$ و $B[3, 5]$ چه خواهد بود؟

(ج) ماتریس A بعد از اجرای $alg1()$ چیست؟

Algorithm 1 $alg1()$

```
1: for  $v = 0$  to  $15$  do
2:    $B[0, v] = 0$ 
3: end for
4: for  $n = 1$  to  $6$  do
5:   for  $v = 0$  to  $15$  do
6:      $A[n, v] = A[n - 1, v]$ 
7:     if  $v \geq V_n$  and  $A[n, v] < A[n - 1, v - V_n] + W_n$  then
8:        $A[n, v] = A[n - 1, v - V_n] + W_n$ 
9:     end if
10:  end for
11: end for
12: return  $A[6, 15]$ 
```

Algorithm 2 $\text{alg2}(n, v)$

```
1: if  $n = 0$  then
2:   return 0
3: end if
4: if  $B[n, v] \neq \text{null}$  then
5:   return  $B[n, v]$ 
6: end if
7:  $B[n, v] = \text{alg2}(n - 1, v)$ 
8: if  $v \geq V_n$  then
9:    $\text{tmp} = \text{alg2}(n - 1, v - V_n) + W_n$ 
10:  if  $B[n, v] < \text{tmp}$  then
11:     $B[n, v] = \text{tmp}$ 
12:  end if
13: end if
14: return  $B[n, v]$ 
```

بخش دوم (۳۵ نمره): با توجه به شماره دانشجویی خود مقدار H و G را محاسبه کنید. اگر باقیمانده $G + H$ بر 2 به ترتیب 0، 1 بود به ترتیب سوال 2 یا 3 را حل کنید.

۲. دو دنباله $\mathbf{a} = (a_1, a_2, \dots, a_n)$ و $\mathbf{b} = (b_1, b_2, \dots, b_m)$ به طول n و m از اعداد صحیح داده شده‌اند. می‌خواهیم تعدادی از اعداد دنباله‌های \mathbf{a} و \mathbf{b} را حذف کنیم که ضرب داخلی دنباله‌های حاصل بیشینه شود. الگوریتمی با زمان اجرای $O(nm)$ برای حل این مسئله طراحی کنید.

تعریف: ضرب داخلی دو دنباله $\mathbf{c} = (c_1, c_2, \dots, c_k)$ و $\mathbf{d} = (d_1, d_2, \dots, d_k)$ برابر $\sum_{i=1}^k c_i \times d_i$ است.

۳. یک وارونگی در جایگشت (a_1, a_2, \dots, a_n) یک زوج مرتب (i, j) است که $i < j$ و $a_i > a_j$. عدد n و k در ورودی داده شده‌اند. الگوریتمی از زمان اجرای $O(nk)$ طراحی کنید که تعداد جایگشت‌هایی به طول n از اعداد 1 تا n را پیدا کند که دقیقاً k وارونگی دارند. در صورتی که زمان اجرای الگوریتم شما $O(n^2k)$ باشد، ۳۰ نمره به شما تعلق خواهد گرفت.

بخش سوم (۴۰ نمره): با توجه به شماره دانشجویی خود مقدار I را محاسبه کنید. اگر باقیمانده I بر 2 به ترتیب 0، 1 بود به ترتیب سوال 4 یا 5 را حل کنید.

۴. یک بازی به این صورت طراحی شده است که شما باید از بین اعداد 1 تا n عدد x را پیدا کنید. شما از ابتدا عدد x را نمی‌دانید و برای پیدا کردن آن می‌توانید تابع isMissing را به صورت زیر اجرا کنید:

- با پرداخت k تومان تابع $\text{isMissing}(k)$ رو صدا کنید. در صورتی که $x = k$ خروجی تابع برابر 0 است. در غیر این صورت اگر $x < k$ باشد خروجی تابع 1- و اگر $x > k$ باشد خروجی تابع 1 است.

هدف پیدا کردن حداقل بودجه مورد نیاز است که تضمین کند که عدد x هر چه باشد آن را پیدا خواهیم کرد. الگوریتم از زمان اجرای $O(n^3)$ برای پیدا کردن حداقل بودجه مورد نیاز طراحی کنید.

مثال: برای مثال برای $n = 3$ حداقل هزینه برابر 2 است. به این ترتیب که با پرداخت 2 تومان تابع $\text{isMissing}(2)$ رو صدا می‌زنیم و با توجه به جواب آن می‌توان عدد x را پیدا کرد.

۵. یک بازی به این صورت طراحی شده است که شما باید از بین اعداد 1 تا n عدد x را پیدا کنید. شما از ابتدا عدد x را نمی‌دانید و برای پیدا کردن آن می‌توانید تابع isMissing را به صورت زیر اجرا کنید:

• می‌توانید تابع $isMissing(k)$ رو صدا کنید. اگر $x \geq k$ خروجی تابع برابر 0 خواهد بود. اگر $x < k$ باشد خروجی تابع برابر 1 خواهد بود و شما باید در این حالت 1 تومان بپردازید.

بودجه شما k تومان است و هدف این است که با کم‌ترین تعداد صدا زدن تابع $isMissing$ عدد x را پیدا کنیم. الگوریتمی با زمان اجرای $O(n^2k)$ طراحی کنید که حداقل تعداد صدا کردن تابع $isMissing$ را پیدا کند که تضمین می‌کند شما در هر حالتی عدد x را پیدا می‌کنید.

مثال ۱: برای مثال اگر $n = 5$ و بودجه 1 تومان باشد جواب مسئله برابر 4 است. به این ترتیب که باید ابتدا $isMissing(2)$ را صدا زد. اگر جواب 1 بود به این معنی است که $x = 1$ است. اگر جواب 0 بود سپس $isMissing(3)$ را صدا می‌کنیم و به همین ترتیب ادامه می‌دهیم.

مثال ۲: برای مثال اگر $n = 4$ و بودجه 2 تومان باشد جواب مسئله برابر 2 است. به این ترتیب که باید ابتدا $isMissing(3)$ را صدا زد. اگر جواب 1 بود به این معنی است که $x < 3$ و با صدا زدن $isMissing(2)$ می‌توان x را پیدا کرد. اگر جواب 0 بود به این معنی است که $x \geq 3$ و با صدا زدن $isMissing(4)$ می‌توان x را پیدا کرد.