

سؤالات تئوری

سطح یک :

1. در یک کارگاه چوب بری، میخواهیم چوبی به طول n متر را به تکه‌های کوچکتر تبدیل کنیم. طول قطعه‌های ایجاد شده عدد صحیحی است و هر کدام ارزش خاصی دارد. الگوریتمی ارائه دهید که مقدار بیشترین ارزش ممکنه از این برش‌ها به دست بیاید.
مثال:: اگر طول چوب برابر 8 باشد و جدول ارزش بر حسب اندازه به شکل زیر باشد

length		1	2	3	4	5	6	7	8

price		1	5	8	9	10	17	17	20

ماکسیمم ارزش برابر 22 خواهد بود (با تقسیم چوب به قطعه‌های 2 متری و 6 متری)
و یا اگر طول چوب برابر 8 باشد و جدول ارزش به شکل

length		1	2	3	4	5	6	7	8

price		3	5	8	9	10	17	17	20

باشد ماکسیمم ارزش برابر 24 خواهد بود (با تقسیم چوب به 8 چوب یک متری)

(راهنمایی : الگوریتم شما باید از $O(n^2)$ باشد)

2. در یک دنباله ی اعداد نامرتب، طول بزرگترین زیردنباله را بیابید که تمام اعضای آن اکیدا صعودی باشد.

مثال: برای دنباله ی $\{10, 22, 9, 33, 21, 50, 41, 60, 80\}$ بزرگترین زیردنباله ای که اکیدا صعودی باشد برابر $\{10, 22, 33, 50, 60, 80\}$ خواهد بود و خروجی الگوریتم شما باید 6 باشد.
همچنین ممکن است این زیردنباله یکتا نباشد و چند زیر دنباله با شرایط بالا وجود داشته باشد که در این صورت پیدا کردن تنها یکی از آنها کافی است.

نکته : الگوریتم شما باید از $O(n^2)$ باشد

3. در یک آرایه ی نامرتب، الگوریتمی پیشنهاد دهید که حاصل جمع بزرگترین زیرآرایه ی متوالی از نظر جمع اعضا را ارائه دهد
مثال : در آرایه ی

$\{-2, -3, 4, -1, -2, 1, 5, -3\}$

بزرگترین زیرآرایه ی متوالی از نظر جمع اعضا برابر

$\{4, -1, -2, 1, 5\}$

است و خروجی الگوریتم شما باید 7 باشد.

همچنین ممکن است این زیر دنباله یکتا نباشد که در این صورت پیدا کردن تنها یکی از آنها کافی است.

نکته : الگوریتم شما باید از $O(n)$ باشد

4. ما در نقطه ی (m, n) قرار داریم (m و n بزرگتر یا مساوی 0 هستند) و میخواهیم به نقطه ی $(0, 0)$ بازگردیم. در هر مرحله میتوانیم یا یک خانه به پایین حرکت کرده یا یک خانه به سمت چپ. یعنی حرکت‌های مجاز ما در هر مرحله $(m-1, n)$ و یا $(m, n-1)$ است.
الگوریتمی ارائه دهید که تعداد حالت‌هایی که میتوانیم با استفاده از حرکت‌های مجاز به خانه ی $0, 0$ بازگردیم را به دست آورد.
نکته : الگوریتم شما باید از $O(n*m)$ باشد.

سطح دو :

1. بر روی یک رشته به طول n ، الگوریتمی ارائه دهید طول بزرگترین زیر رشته ی متقارن آن را به دست آورید.
مثال : برای رشته ی BBABCB CAB خروجی الگوریتم شما باید 7 باشد زیرا BABCBAB بزرگترین زیر رشته ی متقارن آن است. زیر رشته های متقارن دیگری مانند BBBB و BBCBB نیز وجود دارد که بزرگترین نیستند.

نکته : الگوریتم شما باید از $O(n^2)$ باشد

2. در یک ماتریس صفر و یک، بزرگترین زیر ماتریس مربعی را بیابید که تمام عناصر آن یک باشد.

مثال : در ماتریس زیر

1	1	1	0	0
0	1	1	0	0
1	1	1	1	1
0	0	0	0	0

بزرگترین زیر ماتریس مربعی تمام 1 برابر با

1	1
1	1

است. در این حالت خروجی الگوریتم شما باید 2 باشد. (اندازه ی بزرگترین زیر ماتریس)

نکته: الگوریتم شما باید از $O(m*n)$ باشد (m تعداد سطر های ماتریس اصلی و n تعداد ستون ماتریس اصلی است)

3. یک دنباله دو بخشی نامیده می شود اگر ابتدا صعودی و سپس نزولی باشد. یک دنباله ی یک دنباله ی صعودی نیز دو بخشی است (با بخش نزولی به طول صفر) و همچنین یک دنباله ی نزولی نیز دوبخشی حساب میشود.

با داشتن یک دنباله، طول بزرگترین زیر دنباله ی دوبخشی آن را به دست آورید.

نکته : الگوریتم شما باید از $O(n^2)$ باشد.

راهنمایی : میتوانید از الگوریتم خود را که برای «بزرگترین زیر دنباله ی صعودی» ارائه دادید را کمی تغییر دهید تا این مسأله را حل کند.

سطح سه :

1. دو رشته ی A و B داریم و میخواهیم با حذف کردن تعدادی از کاراکتر های A آن را تبدیل به رشته ی B کنیم. تعداد حالت هایی که میتوانیم با حذف کاراکتر های A آن را به B تبدیل کنیم را بیابید.

مثال :

A = "abcccdf", B = "abccdf"

Output : 3

:: Three ways will be -> "ab.ccdf",

"abc.cdf" & "abcc.df" .

"." is where character is removed.

A = "aabba", B = "ab"

4

:: Four ways will be -> "a.b..",

"a..b.", ".ab.." & ".a.b." .

"." is where characters are removed.

2. ضرب ماتریس ها همیشه یک مسأله ی مهم در محاسبات است. فرض کنید n ماتریس دارید که ماتریس I ام دارای ابعاد $k_i * m_i$ است. ما میتوانیم این ماتریس ها را به شکل های مختلف در یکدیگر ضرب کنیم. درواقع ما میتوانیم به شکل های مختلف پرانتز گذاری کنیم. ضرب ماتریسی با ابعاد $m * n$ و ماتریسی با ابعاد $n * k$ دارای هزینه ی $m * n * k$ است. الگوریتمی را پیشنهاد دهید که با انجام پرانتز گذاری مناسب کمترین هزینه ی لازم برای ضرب ماتریس ها را به دست آورد. خروجی الگوریتم شما باید کمترین هزینه ی لازم باشد.

مثال : فرض کنید ضرب ماتریسی زیر وجود دارد:

$$A * B * C$$

که ابعاد ماتریس ها به شکل زیر است

$$A = 3 * 2$$

$$B = 2 * 5$$

$$C = 5 * 4$$

این ضرب را میتوان به دو صورت پرانتز گذاری کرد

$$(A * B) * C$$

$$A * (B * C)$$

که هزینه ی ضرب به شکل اول به اندازه ی ::

$$3 * 2 * 5 + 3 * 5 * 4 = 90$$

و هزینه ی ضرب به شکل دوم به اندازه ی ::

$$3 * 2 * 4 + 2 * 5 * 4 = 64$$

است پس خروجی الگوریتم ما باید عدد 64 باشد.

سؤال های امتیازی :

1. در یک دنباله ی اعداد نامرتب، زیر دنباله ای را زیردنباله ی «خاص» مینامیم که اولاً یک زیردنباله ی متوالی از دنباله ی اصلی باشد و ثانیاً اگر طول این زیردنباله m باشد و مقدار کوچکترین عضو این زیردنباله k باشد مقدار نظیر این زیر دنباله را $m * k$ در نظر میگیریم. بدیهی است که یک دنباله میتواند مقداری زیادی زیر دنباله ی «خاص» داشته باشد. برای یک دنباله ی دلخواه زیردنباله ی «خاصی» را بیابید که مقدار نظیر آن از همه ی زیردنباله های «خاص» دیگر بیشتر باشد.

مثال : دنباله ی $\{1,2,3\}$ دارای زیر دنباله های خاص زیادی است. یکی از آنها $\{1,2\}$ است که مقدار نظیر آن برابر 2 است. یکی دیگر $\{2,3\}$ است که مقدار نظیر آن 4 است. زیر دنباله ی خاص ماکسیمم این دنباله برابر 4 است.
مثال 2 : دنباله ی $\{6, 1, 5, 4, 5, 2, 6\}$ دارای زیردنباله ی خاص $\{5,4,5\}$ است که این زیردنباله ی خاص از تمام زیردنباله های خاص دیگر بزرگتر است.

نکته: الگوریتم شما باید از $O(n)$ باشد

2. در یک ماتریس صفر و یک، بزرگترین زیرماتریس مستطیلی شکلی را بیابید که تمام عناصر آن 1 باشد.

مثال : در ماتریس زیر

$$\begin{matrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{matrix}$$

بزرگترین زیر ماتریس که همه ی عناصر آن 1 باشد برابر

$$\begin{matrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{matrix}$$

است که دارای حاصل 6 می باشد. پس خروجی الگوریتم شما باید 6 باشد.
راهنمایی : میتوانید از قسمت قبل و الگوریتم مربوط به زیرمجموعه ی «خاص» استفاده کنید
نکته : الگوریتم شما باید از $O(n^2)$ باشد

3. تا به حال به صف های پشت پارکینگ ها دقت کرده اید ؟ قرار است یک مسأله به همان شکل حل کنیم. پارکینگی دارای یک ورودی است و در پشت این ورودی یک صف تشکیل شده است. در این صف n ماشین وجود دارد که طول هر کدام میتواند با دیگری متفاوت باشد. در داخل پارکینگ تنها دو لاین وجود دارد. لاین راست و لاین چپ و ماشین ها باید پشت سر هم در یکی از این دو لاین قرار بگیرند. شما در جلوی درب پارکینگ قرار دارید، جایی که قرار است یک صف ماشین در بیرون را به دو صف ماشین (سمت چپ و سمت راست) در داخل پارکینگ تبدیل کنید. طول صف داخل نیز m متر است. یعنی مجموع طول ماشین هایی که در داخل میتوانند در لاین راست قرار بگیرند باید کمتر از m متر باشد.

الگوریتمی پیشنهاد دهید که با تعیین کردن سمت چپ یا راست برای هر ماشین، بیشترین تعداد ماشین را در داخل پارکینگ جا دهد.

دقت کنید که ماشین ها به ترتیب باید وارد صف شوند و اگر ماشین I نتواند وارد پارکینگ شود ماشین $I+1$ نیز نمیتواند وارد پارکینگ شود (خاصیت صف)

مثال : اگر طول پارکینگ ما 50 متر باشد و ماشین هایی به طول 25 متر، 30 متر، 10 متر، 10 متر، 15 متر، 7 متر و 8 متر به همین ترتیب در صف باشد ما میتوانیم به شکل زیر تصمیم گیری کنیم :: ابتدا ماشین 25 متری را وارد صف راست کن. سپس ماشین بعدی (30 متری) را وارد صف چپ کن. سپس ماشین بعدی (10 متری) را وارد صف چپ کن. سپس ماشین بعدی (10 متری) را وارد صف چپ کن. سپس ماشین بعدی (15 متری) را وارد صف راست کن. ماشین بعدی (7 متری) صف راست. ماشین آخر نیز نمیتواند وارد صف شود.

بدین ترتیب در صف سمت راست 47 متر و در صف سمت چپ 50 متر پر شده است. ماشین ها به ترتیب وارد صف شده اند. بیشترین تعداد ماشینی که توانستیم در صف جا دهیم نیز 6 ماشین بوده است.

سؤالات عملی

- 1.
- 2.
- 3.
- 4.
- 5.

سؤالات حل TA:

1. 8 وزیر
2. Tilling problem
3. Independent Set بر روی درخت
4. بزرگترین زیر دنباله ی زیگ زاگی