بخش اول:

۱. در این سوال هدف اجرای الگوریتم ضرب چند جمله‌ای با زمان اجرای $O(n^{\log_2^3})$ است. با توجه به شماره دانشجویی خود مقادیر G، H و I را محاسبه کنید و فرض کنید می‌خواهیم حاصل ضرب دو چند جمله‌ای $A(x) = 1 + G \times x + x^2 + 3 \times x^3$ و $B(x) = 3 - I \times x^2 + H \times x^3$ را حساب کنید

- به طور دقیق زیر مسئله‌های مرتبط چه زیر مسئله‌هایی هستند؟
- جواب هر زیر مسئله را مشخص کنید
- جواب نهایی را از روی جواب زیر مسئله‌ها حساب کنید. مراحل رسیدن به جواب نهایی رو مشخص کنید.

راه حل:

سوال مشابه در تمرین:

۱. همانطور که در کلاس بحث شد، الگوریتم به این صورت می‌باشد:

$$A(x) = 3 + 2x + 5x^2 + x^3 = A_0(x) + A_1(x)x^2 \Rightarrow A_0(x) = 3 + 2x, \quad A_1(x) = 5 + x$$
$$B(x) = 1 + 3x + 4x^3 = B_0(x) + B_1(x)x^2 \Rightarrow B_0(x) = 1 + 3x \quad B_1(x) = 4x$$
$$AB = A_0B_0 + [(A_0 + A_1)(B_0 + B_1) - A_0B_0 - A_1B_1]x^2 + A_1B_1x^4$$
$$A_0B_0 = 3 \times 1 + [(3 + 2)(1 + 3) - 3 \times 1 - 2 \times 3]x^1 + (2 \times 3)x^2 = 3 + 11x + 6x^2$$
$$A_1B_1 = 5 \times 0 + [(5 + 1)(0 + 4) - 5 \times 0 - 1 \times 4]x^1 + (1 \times 4)x^2 = 20x + 4x^2$$
$$(A_0 + A_1)(B_0 + B_1) = (8 + 3x)(1 + 7x) = 8 \times 1 + [(8 + 3)(1 + 7) - 8 \times 1 - 3 \times 7]x^1 + (3 \times 7)x^2$$
$$\Rightarrow (A_0 + A_1)(B_0 + B_1) = 8 + 59x + 21x^2$$
$$\Rightarrow AB = 3 + 11x + 6x^2 + [8 + 59x + 21x^2 - (3 + 11x + 6x^2) - (20x + 4x^2)]x^2 + (20x + 4x^2)x^4$$
$$\Rightarrow AB = 3 + 11x + 6x^2 + [5 + 28x + 11x^2]x^2 + (20x + 4x^2)x^4$$
$$\Rightarrow AB = 3 + 11x + 11x^2 + 28x^3 + 11x^4 + 20x^5 + 4x^6$$

بخش دوم:

۲. درخت دودویی درختی است که دارای خواص زیر است:

- به هر هر راس یک عدد نسبت داده شده است.
- به غیر از ریشه، هر راس یک پدر دارد. همچنین هر راس حداکثر دو فرزند دارد که آنها را فرزند سمت چپ و راست می‌نامیم.

درخت جستجوی دودویی یک درخت دودویی است که خاصیت زیر را نیز دارد:

- به ازای هر راس، عدد نسبت داده شده به آن از تمام اعداد زیردرخت سمت چپ آن بزرگتر است و از تمام اعداد زیر درخت سمت راست آن بزرگتر نیست.

یک درخت دودویی به n راس داده شده است. الگوریتمی با زمان اجرای $O(n)$ طراحی کنید که تشخیص دهد که آیا درخت ورودی یک درخت جستجویی دودویی است. شبه کد مربوط به الگوریتم خود را بنویسید. در صورتی که زمان اجرای الگوریتم شما $O(n \log n)$ باشد قسمتی از نمره به شما تعلق می‌گیرد.

راه حل:

لینک

۳. یک آرایه به طول n از اعداد 0 تا n داده شده است. می‌دانیم در این آرایه تمام اعداد به جز یک عدد آمده است که آن را عدد گمشده می‌نامیم. الگوریتم با زمان اجرای $O(n)$ و حافظه کمکی $O(\log n)$ طراحی کنید که عدد گمشده را پیدا کند. شبه کد مربوط به الگوریتم خود را بنویسید.

راه حل:

۱- عدد n/2 را به عنوان pivot انتخاب می کنیم. اعداد را مانند الگوریتم quick sort پارتیشن می کنیم. از روی سایز دو دسته می توان فهمید عدد گمشده در کدام دسته است. دسته دیگر را کنار می گذاریم و عملیات را روی دسته دیگر به صورت بازگشتی ادامه می دهیم. مرتبه زمانی است. $n + n/2 + n/4 + \ldots = O(n)$

۲- جمع اعداد آرایه را محاسبه می کنیم. جمع اعداد ۰ تا n یا $\frac{n(n+1)}{2}$ را از مجموع اعداد کم می کنیم، حاصل برابر عدد گمشده است.

بخش سوم:

۴. رئوس یک چند ضلعی محدب در مختصات دکارتی (x, y) داده شده است. می‌دانیم هیچ دو راس این چند ضلعی مختصات x یا y یکسان ندارند. رئوس چند ضلعی به ترتیب ساعت‌گرد با شروع از سمت چپ‌ترین راس (راس با کمترین x) داده شده‌اند. الگوریتمی با زمان اجرای $O(\log n)$ طراحی کنید که بالاترین راس (راس با بیشترین y) و سمت راست‌ترین راس (راس با بیشترین x) رای پیدا کند.

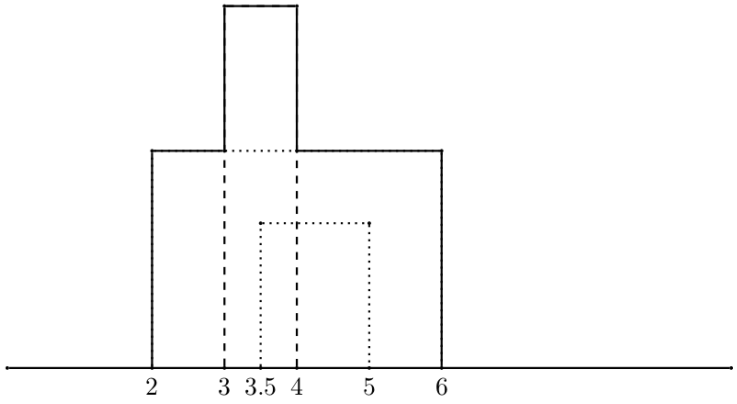
راه حل:

۵. یک آرایه را unimodal می نامیم هر گاه از ابتدای آرایه تا جایی اعداد به صورت صعودی باشند و از آنجا به بعد به صورت نزولی. حال ادعا می کنیم این تعییر جهت در چنین آرایه ای را می توان در O(lgn) بدست آورد. ابتدا عنصر میانی را نگاه می کنیم و اگر از عدد قبل بزرگتر. و از عدد بعد کوچکتر باشد در قسمت صعدی قرار دارد و می توانیم تغییر را در نیمه دوم آرایه که آن هم unimodal است جستحو کنیم، همین طور اگر از عدد قبل کوچکتر و از عدد بعد بزرگتر بود در قسمت نزولی قرار دارد و جواب در نیمه اول آرایه است. و اگر از عنصر بعد بزرگتر و از عنصر قبل هم بزرگتر باشد، عنصر مورد نظر می باشد.
به همین صورت مانند جست و جوی دودویی ادامه می دهیم تا این عنصر را پیدا کنیم. حال توجه می کنیم که این نقاط بر حسب مولفه x، unimodal هستند و نقطه با بیشترین x هم همان نقطه تغییر جهت است. پس. در O(lgn) می شود آن را پیدا کرد. سپس اگر از این نقطه تا انتهای نقطه ها به آرایه نگاه کنیم بر حسب مولفه y، unimodal است و عنصر با بیشترین y نقطه تغییر جهت است و آن را می توان در O(lgn) پیدا کرد.

اشتباه راه حل: در آخر راه حل باید از اول تا نقطه با بیشترین x را در نظر بگیریم تا بتوانیم بیشترین y را پیدا کنیم.

۵. روی یک صفحه n مستطیل سیاه کشیده‌ایم. ضلع پایینی تمامی مستطیل‌ها روی محور x (خط $y = 0$) قرار دارند. مستطیل iـام با سه‌تایی (l_i, r_i, h_i) توصیف می‌شود به این معنی که l_i و r_i به ترتیب مختصات ضلع سمت چپ و راست مستطیل روی محور x و h_i ارتفاع آن است. هدف این است که شکل نهایی از قرار دادن این مستطیل‌ها روی صفحه را پیدا کنیم. به بیانی دیگر شکل نهایی حاصل قرار گرفتن این مستطیل‌های سیاه یک شکل پلکانی‌ـمانند می‌شود. باید الگوریتمی طراحی کنید که خروجی پلکانی‌ـمانند را تولید کند و برای نقاطی که تغییر ارتفاع صورت می‌گیرد مختصات x و ارتفاع جدید رو در خروجی چاپ کنید. الگوریتمی با زمان اجرای $O(n \log n)$ برای این مسئله طراحی کنید.

- برای مثال فرض کنید در ورودی سه مستطیل با سه‌تایی‌های $(2, 6, 3)$، $(3.5, 5, 2)$ و $(3, 4, 5)$ داده شده‌اند. خروجی نهایی $((2, 3), (3, 5), (4, 3), (6, 0))$ خواهد بود به این دلیل که ابتدا در $x = 2$ ارتفاع شکل حاصل 3 می‌شود، سپس در $x = 3$ به خاطر مستطیل سوم ارتفاع 5 می‌شود، در $x = 4$ ارتفاع به 3 کاهش پیدا می‌کند و در نهای در $x = 6$ ارتفاع 0 خواهد شد.



راه حل:

لینک