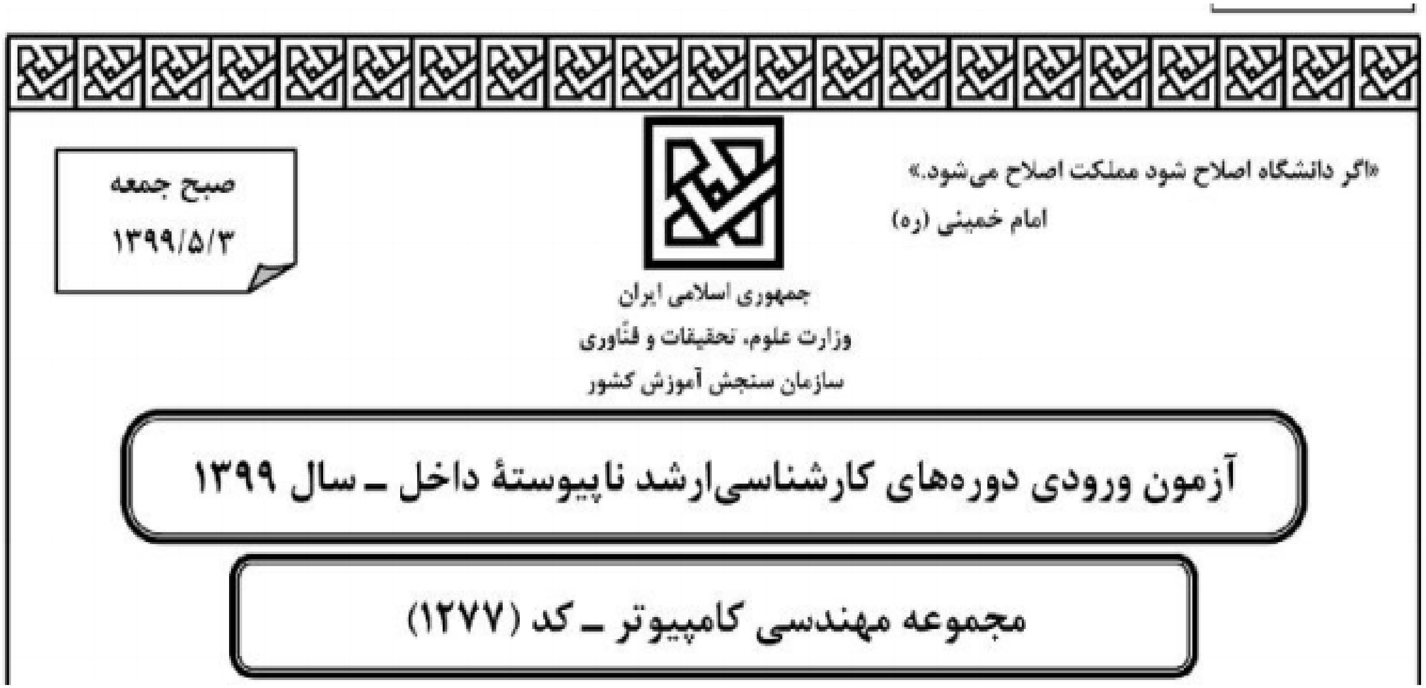
بسم الله الرحمن الرحیم

پاسخ سوالات

ساختمان داده و طراحی الگوریتم

در آزمون ارشد ۹۹



«اگر دانشگاه اصلاح شود مملکت اصلاح می‌شود.»
امام خمینی (ره)

صبح جمعه
۱۳۹۹/۵/۳

جمهوری اسلامی ایران
وزارت علوم، تحقیقات و فنّاوری
سازمان سنجش آموزش کشور

آزمون ورودی دوره‌های کارشناسی‌ارشد ناپیوستهٔ داخل ـ سال ۱۳۹۹

مجموعه مهندسی کامپیوتر ـ کد (۱۲۷۷)

مهندسی کامپیوتر

ابوالفضل گیلک

گروه بابان

@abolfazlgilak

۸۵- به ازای چند زوج (a , b) از اعداد طبیعی کوچکتر از ۵، جواب رابطهٔ بازگشتی $T(n) = aT(n/b) + n^{2}$ برابر $\theta(n^{2})$ می‌شود؟

۱) ۷

۲) ۸

۳) ۱۱

۴) ۱۲

پاسخ: گزینه (۳)

$p = \log_b a$

$n^p \quad n^2$

$p < 2 \implies \log_b a < 2 \implies a < b^2$

$b = 1 \implies$ هیچ مقدار a

$b = 2 \implies a = 1, 2, 3$

$b = 3 \implies a = 1, 2, 3, 4$

$b = 4 \implies a = 1, 2, 3, 4$

در مجموع ۱۱ حالت برای (a,b) داریم.

۸۶- جدول درهم‌سازی ۱۰ خانه‌ای و تابع درهم‌ساز $h(x) = ۳x + ۵ \bmod ۱۰$ و روش زنجیره‌ای به‌عنوان روش رفع تصادم را در نظر بگیرید. در این خصوص کدام گزینه درست است؟

۱) احتمال آن‌که ورودی $x = ۴$ به خانه ۷ نگاشت شود برابر ۱ است.

۲) احتمال آن‌که ورودی $x = ۴$ به خانه ۷ نگاشت شود برابر ۰ است.

۳) احتمال آن‌که ورودی $x = ۴$ به خانه ۷ نگاشت شود برابر $\frac{۱}{۱۰}$ است.

۴) احتمال آن‌که دو ورودی مختلف به یک خانه نگاشت شوند برابر $\frac{۱}{۱۰}$ است.

پاسخ: گزینه (۱)

در روش زنجیره‌ای برای هر x، آدرس $h(x)$ همان آدرس درج x است:

$$x = ۴ \implies h(۴) = ۱۷ \bmod ۱۰ = ۷$$

توجه: اگر از آدرس دهی باز استفاده می‌شد و h یکنواخت بود، گزینه‌های ۳ و ۴ می‌توانست درست باشد.

۸۷- فرض کنید یک آرایهٔ مرتب از n عدد صحیح در اختیار داریم. با چه تعداد مقایسه می‌توانیم بفهمیم عددی بیش از n/۵ بار در آرایه تکرار شده است یا خیر؟ (بهترین گزینه را انتخاب کنید.)

۱) $O(n)$

۲) $O(\log n)$

۳) $O(\log^* n)$

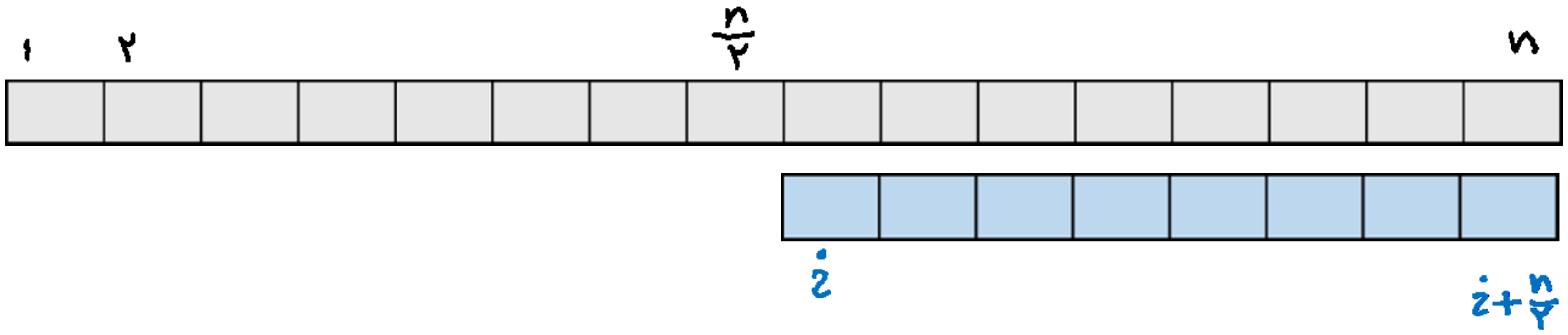
۴) $O(\log\log n)$

پاسخ: گزینه ۲)

ابتدا مسئله را برای $\frac{n}{2}$ حل می‌کنیم. نشان می‌دهیم در زمان $O(\log n)$ می‌توان در یک لیست مرتب از اعداد صحیح، عددی که حداقل $\frac{n}{2}$ بار تکرار شده را یافت.

ما به دنبال قطعه‌ای به طول $\frac{n}{2}$ هستیم که ابتدا و انتهای آن برابر باشد. یعنی $A[i+\frac{n}{2}-1] = A[i]$.

هدف ما یافتن ابتدا (i) یا انتهای ($i+\frac{n}{2}-1$) این قطعه است.



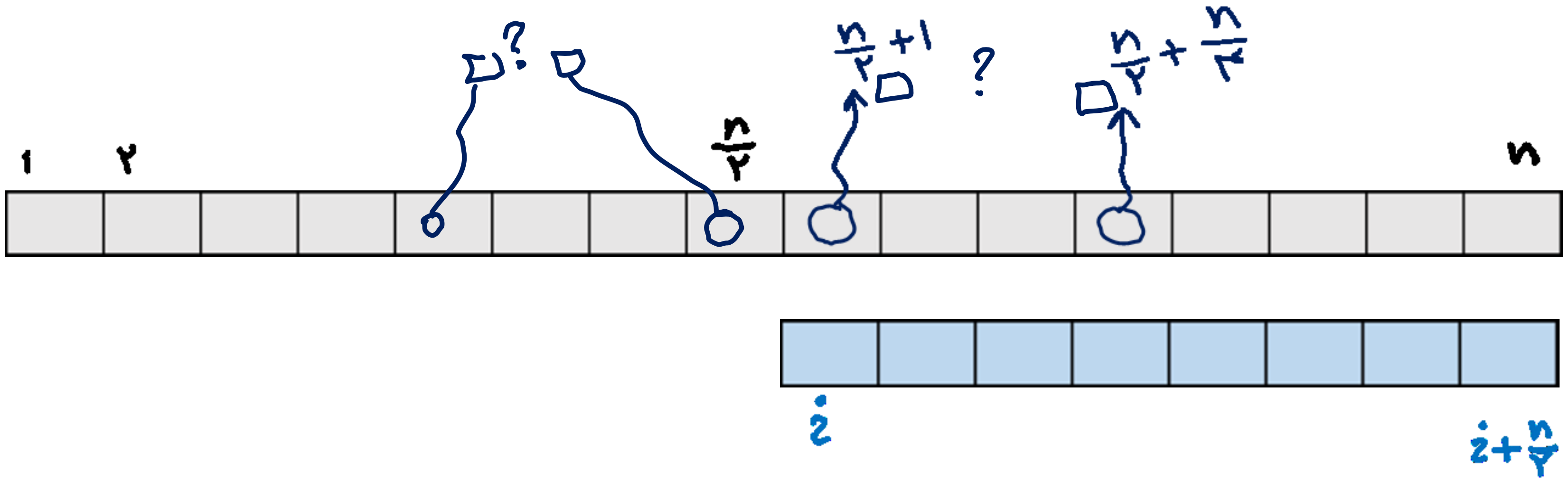
در شروع کار می‌دانیم که:

$$i \in \{1, 2, \dots, \tfrac{n}{2}+1\}$$

$$i+\tfrac{n}{2}-1 \in \{\tfrac{n}{2}, \dots, n\}$$

یعنی اندازه مسئله ما $\frac{n}{2}$ است. اما با $O(c)$ مقایسه می‌توان مقادیر ممکن i را نصف کرد:

با نصف کردن طول این قطعه، در نظر گرفتن قطعه به طول $\frac{n}{4}$. مقایسه‌ی $A[\frac{n}{2}+1]$ و $A[\frac{n}{2}+\frac{n}{4}]$ داریم:



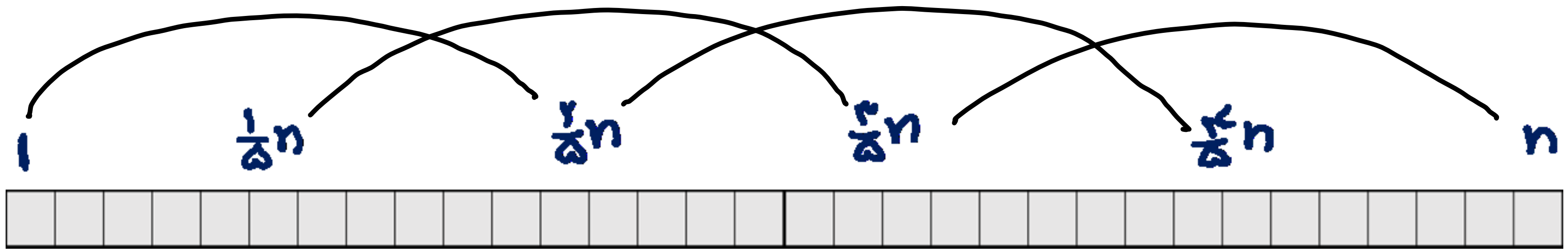
$$A\left[\frac{n}{2}+1\right] \neq A\left[\frac{n}{2}+\frac{n}{4}\right] \implies i \in \left\{1, 2, \dots, \frac{n}{4}\right\}$$

$$A\left[\frac{n}{2}\right] \neq A\left[\frac{n}{2}-\frac{n}{4}+1\right] \implies i+\frac{n}{2}-1 \in \left\{\frac{n}{2}+\frac{n}{4}-1, \dots, n\right\}$$

بنابراین با حداکثر ۲ مقایسه، مقادیر ممکن برای x نصف می‌شود. پس مسئله در $O(\log n)$ حل می‌شود.

حالا برای مسئله‌ی مطرح شده در این سوال، همان الگوریتم قبل را برای بازه‌های به طول $N = \frac{2}{5}n$ مورد استفاده قرار دهید. دقت کنید

$$\text{if} \quad N = \frac{2}{5}n \implies \frac{1}{2}N = \frac{n}{5}$$



باید الگوریتم قبلی را برای ۴ قطعه‌ی $A[k, k+\frac{2}{5}n]$ به ازای $k=1, \frac{n}{5}, \frac{2n}{5}, \frac{3n}{5}$ تکرار کنیم. می‌دانیم که

$$4 \times \log n = O(\log n)$$

۸۸- **اعداد صحیح $x_1, \dots, x_n$ را در یک درخت دودویی جستجو با ارتفاع h ذخیره کرده‌ایم. فرض کنید هزینهٔ جستجوی x_i (تعداد مقایسه‌های لازم در درخت برای پیدا کردن x_i) برابر c_i باشد. می‌دانیم $\sum_{i=1}^{n} c_i = O(n \log n)$ است. کدام گزینه زیر درست است؟**

۱) $h = \Omega(\sqrt{n})$

۲) $h = O(\log n)$

۳) $h = O(\sqrt{n \log n})$

۴) می‌توان مثالی زد که $h = \Omega(n)$ باشد.

پاسخ: گزینه (۳)

برای جستجوی کلید (جستجوی موفق) می‌دانیم که:

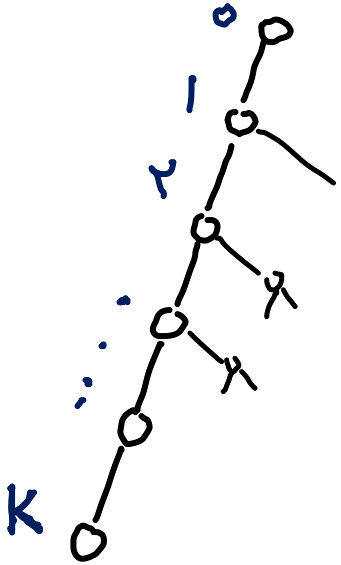
$c_i = 1 + d_i$

$$S = \text{مجموع تعداد مقایسه‌ها} = \sum_{i=1}^{n} \overbrace{(d_i + 1)}^{c_i} = n + \sum_{i=1}^{n} d_i$$

و چون عمق همه گره‌ها به حداقل، حداقل یک است پس می‌توان گفت: $S = \theta(\sum d_i)$

که d_i عمق گره i ام است. پس طبق صورت سوال $\sum_{i=1}^{n} d_i = O(n \log n)$ و

از اینجا می‌خواهیم حدود ارتفاع را تعیین کنیم.



دقت کنید اگر طولانی ترین مسیر از ریشه تا برگ ، از ریشه تا برگ به عمق

K باشه آنگاه حداقل روی آن مسیر داریم :

$$\sum d_i = 0 + 1 + 2 + \cdots + K = \frac{K(K+1)}{2} = \Theta(K^2)$$

(برای گره‌های این مسیر)

طبق صورت سوال $\Longrightarrow$ $K^2 \in O(n \log n)$ $\Rightarrow$ $K \in O(\sqrt{n \log n})$

پس باید این حداکثر عمق برگ‌ها که همان ارتفاع درخت است باید $O(\sqrt{n \log n})$ باشه.

دقت کنید که طبق همین محاسبه گزینه‌های (۱) و (۴) نادرست هستند زیرا اگر ارتفاع درخت $h=\Theta(n)$ باشد آنگاه $\sum d_i = \Omega(n^2)$ خواهد بود. همچنین گزینه (۳) در جهت عکس این حکم درست است یعنی:

$$\text{if} \qquad h = O(\log n) \implies \sum d_i = O(n\log n)$$

اما عکس این مطلب درست نیست. دیدیم که:

$$\text{if} \qquad \sum d_i = O(n\log n) \implies h = O(\sqrt{n\log n})$$

۸۹- فرض کنید $T(n)$ متوسط زمان اجرای الگوریتم مرتب‌سازی سریع به ازای همه جایگشت‌های ممکن ورودی از n عدد متمایز باشد. کدام رابطهٔ بازگشتی زیر درست است؟

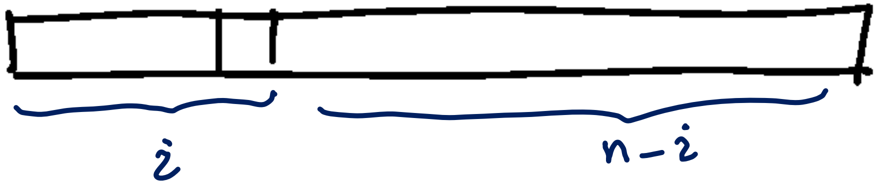
۱) $T(n) = O(n) + \sum_{i=0}^{n-1} \left(T(i) + T(n-i) \right)$

۲) $T(n) = O(n) + (1/n) \sum_{i=0}^{n-1} \left(T(i) + T(n-i) \right)$

۳) $T(n) = (1/n)\left(O(n) + \sum_{i=0}^{n-1} \left(T(i) + T(n-i) \right)\right)$

۴) $T(n) = O(n) + (1/n) \sum_{i=0}^{n-1} \min\left(T(i) , T(n-i) \right)$

پاسخ، گزینه (۲)



در یک جایگشت تصادفی (random) با احتمال یکسان برای همه‌ی جایگشت‌ها داریم :

$$T(n) = \sum_{i=0}^{n-1} \frac{1}{n} \left(O(n) + T(i) + T(n-i) \right)$$

احتمال آن که عنصر محور انتخاب شده آماره‌ی ترتیبی $i+1$ اُم باشد.

زمان لازم برای افراز داده‌ها

مرتب کردن سمت چپ

مرتب کردن سمت راست.

البته دقت کنید که استفاده از $T(i) + T(n-i-1)$ دقیق‌تر بود. اما به لحاظ مرتبه، تفاوتی

ایجاد نمی‌شه. با مرتب کردن جملات و خارج کردن $\frac{1}{n}$ از سیگما و دقت به این‌که $\sum_{i=0}^{n-1} O(n) = n\,O(n)$

به گزینه (۲) می‌رسیم.

۹۰- فرض کنید T درخت جستجوی عمق اول گراف همبند و بدون جهت G است. دو رأس u و v در این درخت برگ و در G دارای درجه حداقل ۲ هستند. کدام‌یک از گزاره‌های زیر صحیح است؟

الف) باید یک رأس w وجود داشته باشد که با u و v در G همسایه باشد.

ب) باید یک رأس w وجود داشته باشد که حذف آن u را از v در G جدا می‌کند.

۱) (الف) نادرست، (ب) نادرست

۲) (الف) نادرست، (ب) درست

۳) (الف) درست، (ب) نادرست

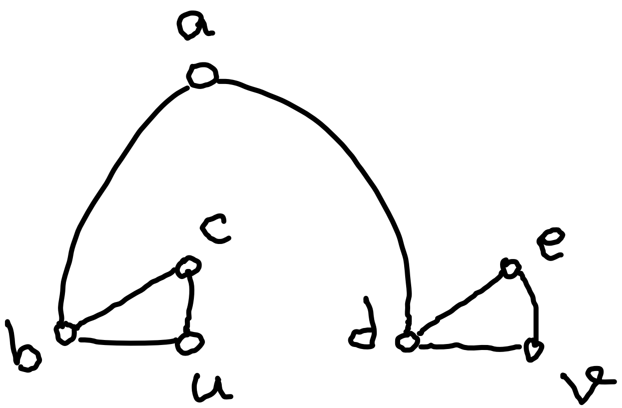
۴) (الف) درست، (ب) درست

پاسخ: گزینه (۱)

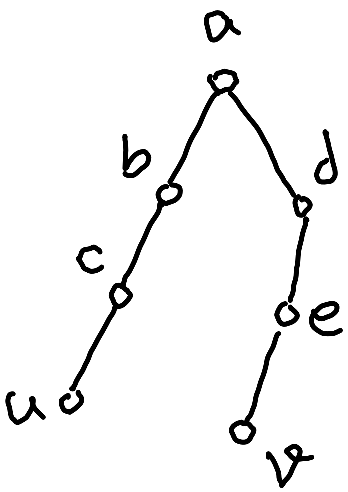
هرکدام از این ادعاها با یک مثال نقض رد می‌شوند. [نحوه رسیدن به این مثال‌ها را می‌توانید در فیلم حل سوالات ارشد ۹۹ در کانال و سایت بابان و آپارات ببینید].

مثال نقض برای رد کردن (الف)

G:

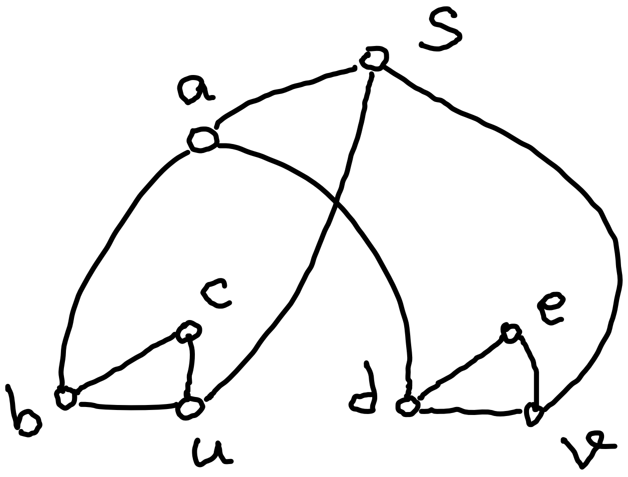


DFS:

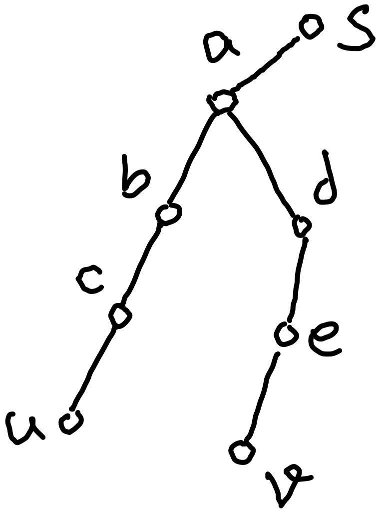


مثال نقض برای رد کردن (ب)

G:



DFS:



۹۱- فرض کنید آرایه $A[1..n]$ شامل n عدد صحیح متمایز است که به صورت صعودی مرتب شده‌اند. چند تا از مسائل زیر را می‌توان با مرتبهٔ زمانی $O(\log n)$ حل کرد؟

- پیدا کردن یک اندیس i طوری‌که $A[i] = i$ شود.
- پیدا کردن یک اندیس i طوری‌که $A[i] = 3i + 2$ شود.
- پیدا کردن یک اندیس i طوری‌که $A[i] = 4i^2 + 3i + 5$ شود.

۱) ۰

۲) ۱

۳) ۲

۴) ۳

پاسخ: گزینه (۲)

می‌خواهیم مسأله را در یک دیدگاه کلی‌تر از درس ساختمان داده مورد بررسی قرار دهیم. ما می‌توانیم آرایه‌ی $A[1,2,\dots,n]$ را به صورت یک تابع تصور کنیم که $f(i)=A[i]$ است.

در درس ریاضی عمومی، وقتی از شیب منحنی صحبت می‌کنیم منظورمان $\dfrac{\Delta y}{\Delta x}$ است و البته با کمک مشتق‌گیری هم می‌توان شیب را به دست آورد. برای مثال اگر $A[i]=2i$ باشد، و آن را مانند تابع $f(x)=2x$ تصور کنیم، شیب آن $f'(x)=2$ است. در واقع وقتی به دو نقطه‌ی متوالی $(i,2i)$ و $(i+1,2(i+1))$ نگاه می‌کنیم $\dfrac{\Delta y}{\Delta x}=\dfrac{2}{1}=2$ است.

ابتدا به توابع حقیقی f و g در حالت کلی توجه کنید: (شکل های اسلاید بعدی)

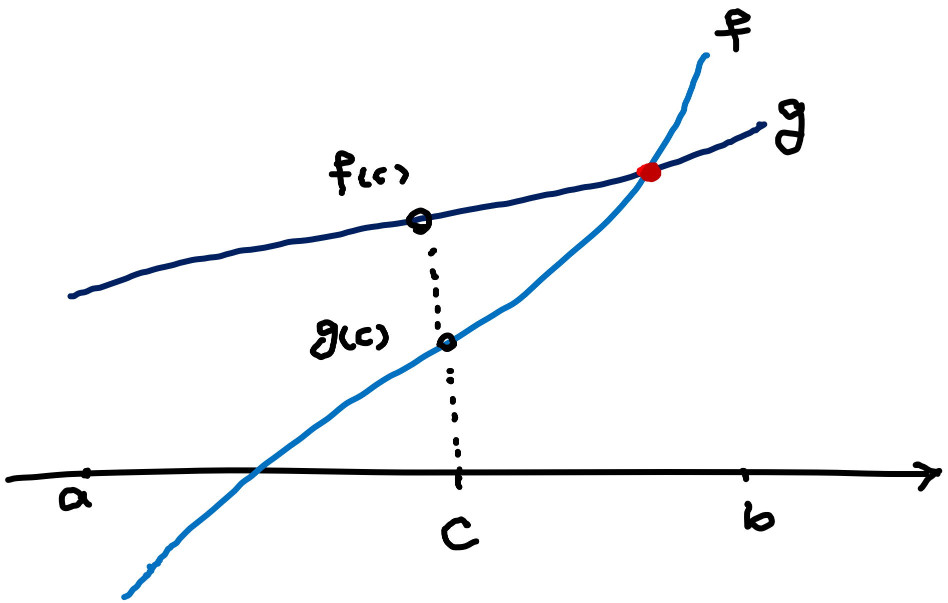
اگر دو تابع حقیقی و صعودی f و g داشته باشیم و شیب g، کمتر از شیب f باشد و به دنبال محل برخورد احتمالی آنها باشیم، ابتدا مقادیر f و g را در نقطه ای مانند c مقایسه می کنیم

$$c = \frac{a+b}{2}$$

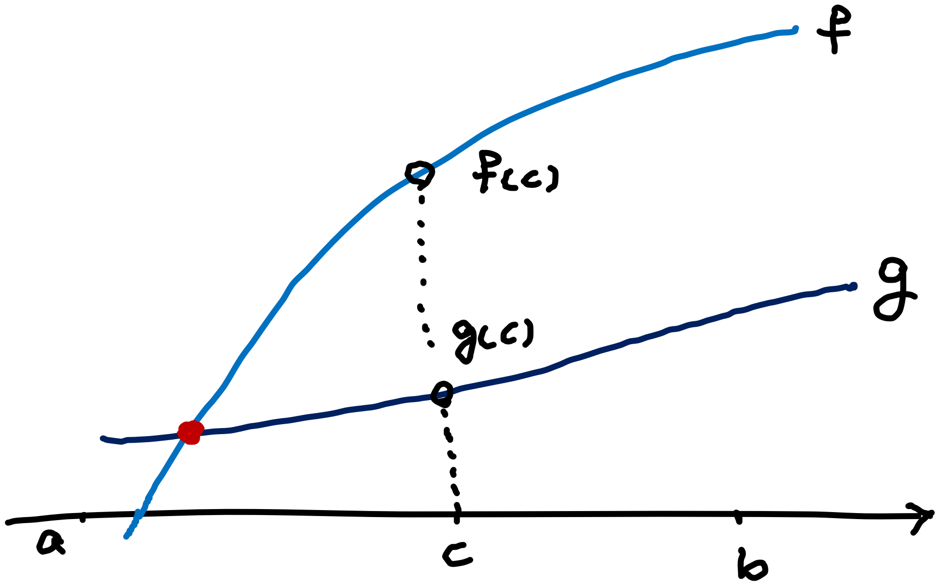
if $f(c) > g(c)$ $\longrightarrow$ محل برخورد احتمالی در سمت چپ c است

if $f(c) < g(c)$ $\longrightarrow$ محل برخورد احتمالی، در سمت راست c است.

پس به شرط آن که شیب یک تابع همواره بیشتر (مساوی) دیگری باشد، می توان از جستجوی دودویی در زمان $O(\log n)$ استفاده کرد.

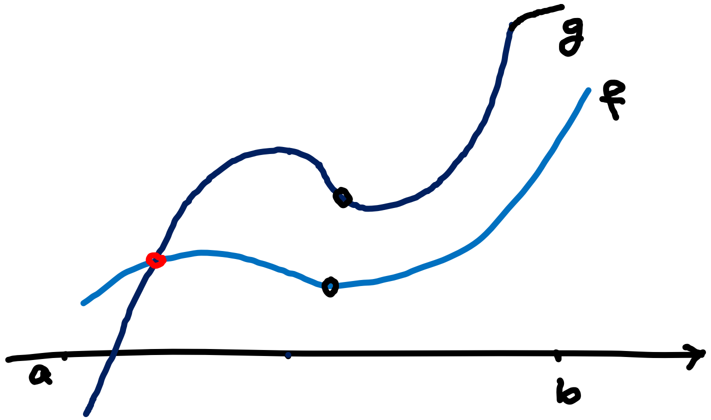


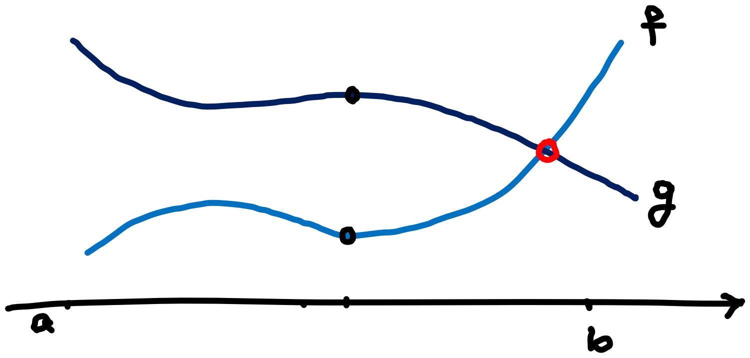
$f(c) > g(c) \Rightarrow$ نقطه برخورد سمت راست c قرار دارد.



$f(c) < g(c) \Rightarrow$ نقطه برخورد سمت چپ c قرار دارد.

اما اگر شیب f و شیب g رابطه معناداری نداشته باشند، از این که مثلاً $f(c) < g(c)$ شده است هیچ نتیجه‌ای نمی‌توان گرفت که جواب، در چپ قرار دارد یا در راست. (به این نمودارها دقت کنید:)





نتیجه: شرط آن که بتوانیم با هر مقایسه، دامنه را نصف کنیم و از جستجوی دودویی در زمان $O(\log n)$ استفاده کنیم آن است که شیب یک منحنی همواره بیشتر مساوی دیگری باشد.

$A[i]$ از اعداد صحیح متمایز و صعودی تشکیل شده است.

پس $A[i+1] - A[i] \geq 1$

در واقع به عنوان یک تابع، شیب آن حداقل یک است. (شیب $A[i] \geq 1$)

$g(i) = i$ شیب دقیقاً 1 است.

$g(i) = 3i + 2$ شیب دقیقاً 3 است.

$g(i) = 4i^2 + 3i + 5$ شیب آن $4i + 3$ متغیر است. (بستگی به مقدار i دارد)

فقط در $g(i) = i$ مطمئن هستیم که شیب $A[i]$ همیشه بیشتر مساوی $g(i)$ است.

۹۲- **چند مورد از مسائل ان‌پی ـ سخت زیر هنگامی که ورودی یک درخت بدون وزن است، در زمان چند جمله‌ای قابل حل هستند؟**

- **پیدا کردن طولانی‌ترین مسیر**
- **محاسبه کمینه رنگ مورد نیاز برای رنگ‌آمیزی رأسی به‌طوری‌که رئوس مجاور همرنگ نباشند.**
- **پیدا کردن بزرگ‌ترین مجموعهٔ مستقل (مجموعه رئوسی که بین هر دو رأس یالی وجود نداشته باشد.)**

۱) ۰

۲) ۱

۳) ۲

۴) ۳

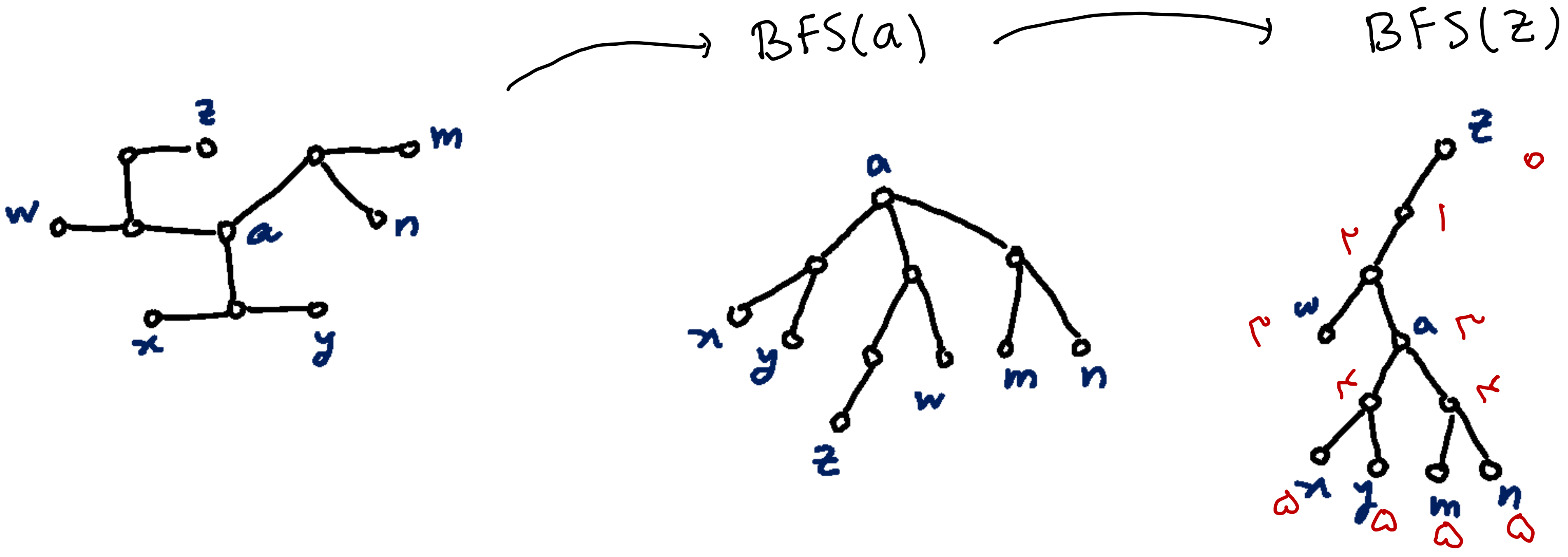
پاسخ: گزینه (۴)

همه‌ی این مسائل، روی درخت‌ها راه حل از مرتبه‌ی خطی دارند و هر ۳ آنها در دوره حل تمرین ۵۰٪ مسأله حل شده‌اند.

الف: پیدا کردن طولانی‌ترین مسیر در یک درخت:

اولاً: در درخت بین دو جفت از گره‌ها دقیقاً یک مسیر وجود دارد که همان طولانی‌ترین مسیر بین آن جفت از گره‌ها است و توسط BFS(n) به دست می‌آید.

ثانیاً: اگر منظور سؤال طولانی‌ترین مسیر در کل درخت باشد همان قطر درخت است که با دوبار اجرای DFS به دست می‌آید.



diam(G) = ۵

مرحله اول: اجرای DFS(a) از یک گره دلخواه. محاسبه ی $d(v)$ برای همه گره ها $O(n)$

مرحله دوم: در نظر گرفتن گره ی با بیشترین عمق $O(n)$

مرحله سوم: اجرای DFS از آن گره. $O(n)$

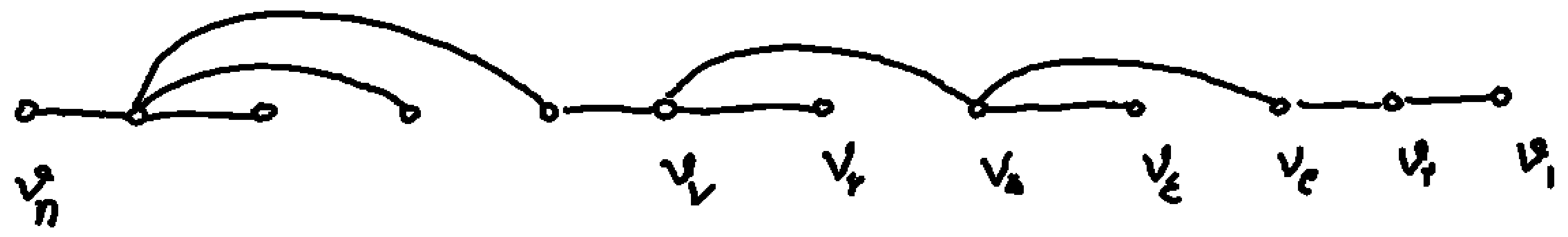
مرحله چهارم: محاسبه ی حداکثر عمق گره ها در این درخت DFS. (جواب، قطر درخت است) $O(n)$

ب) درخت‌های غیر بدیهی دو بخشی هستند و عدد رنگی آنها ۲ است. ضمن آن که نیازی به اجرای الگوریتم نداریم زیرا از قبل جواب را می‌دانیم، اگر هم بخواهیم می‌توانیم دو بخشی بودن همه گراف‌ها (از جمله درخت‌ها) را با اجرای BFS و مقایسه‌ی شماره‌ی سطح گره‌های مجاور برای هر یال تشخیص دهیم. این کار برای تمام گراف‌های دو بخشی حتی اگر درخت هم نباشند در زمان $O(V+E)$ انجام می‌شود.

(*) به صورت کلی تشخیص دو بخشی بودن هر گراف، در $O(V+E)$ انجام می‌شود. درخت‌ها هم دو بخشی هستند. هر گراف دو بخشی غیر تهی هم عدد رنگی‌اش ۲ است.

ج) یکی از مسائل معروف که به روش پویا روی درخت ۲ در زمان چندجمله‌ای حل می‌شود. به توضیحات توجه کنید.

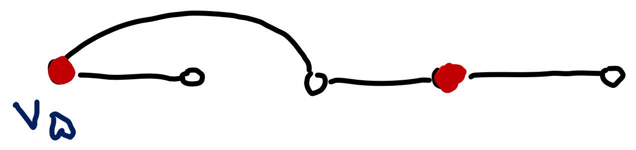
ابتدا مانند مساله‌ی قبل در زمان $O(n)$ ترتیب توپولوژیک ایجاد می‌کنیم.

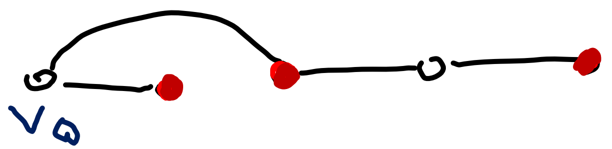


حال مثلاً برای v_5 داریم:

$A_5 = 2$

$B_5 = 3$





ابتدا در زمان $O(n)$ ترتیب توپولوژیکی به گره‌ها می‌دهیم. پس برای هر گره v برای زیردرخت به ریشه‌ی v مقادیر A_v و B_v را تعریف می‌کنیم. هر دوی آنها بزرگترین مجموعه مستقل رئوس‌ها با ریشه‌ی v هستند تنها تفاوت در آنها این است که $v \in A_v$ و $v \notin B_v$.

البته در نهایت $P_v = \max(A_v, B_v)$ جواب زیرمسأله‌ی v است. اما روش پویا به این صورت پیش می‌رود که

$$A_v = 1 + \sum_{u \in child(v)} B_u \qquad B_v = \sum_{u \in child(v)} A_u$$

۹۳- گراف همبند و وزن‌دار G را در نظر بگیرید. درخت پوشای کمینه G لزوماً یکتا نیست و G می‌تواند چندین درخت پوشای کمینه داشته باشد. فرض کنید T یک درخت پوشای کمینه دلخواه از G باشد. کدام‌یک از گزاره‌هـای زیـر درست هستند؟

(الف) حتماً یک اجرا از الگوریتم پریم وجود دارد که T را تولید کند.

(ب) حتماً یک اجرا از الگوریتم کروسکال وجود دارد که T را تولید کند.

توجه کنید زمانی‌که وزن یال‌ها متمایز نیست، می‌توان اجراهای متفاوتی برای هر دو الگوریتم پریم و کروسکال در نظر گرفت. در واقع وقتی چندین یال دارای وزن یکسان باشند، می‌توان هر ترتیبی از آن‌ها را برای پردازش مـورد نظر الگوریتم‌های فوق در نظر گرفت.

۱) (الف) درست، (ب) درست

۲) (الف) درست، (ب) نادرست

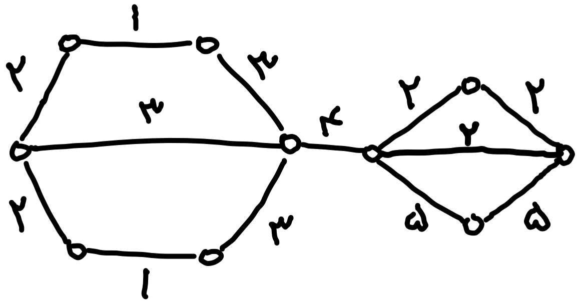
۳) (الف) نادرست، (ب) درست

۴) (الف) نادرست، (ب) نادرست

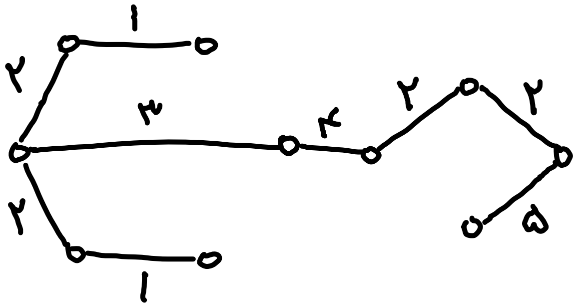
پاسخ: گزینه (۱)

هر درخت MST که توسط کراسکال تولید شود، توسط پریم نیز به دست می‌آید.

فقط باید نشان دهیم هر درخت MST می‌تواند نتیجه‌ی نهایی (خروجی) کراسکال باشد.

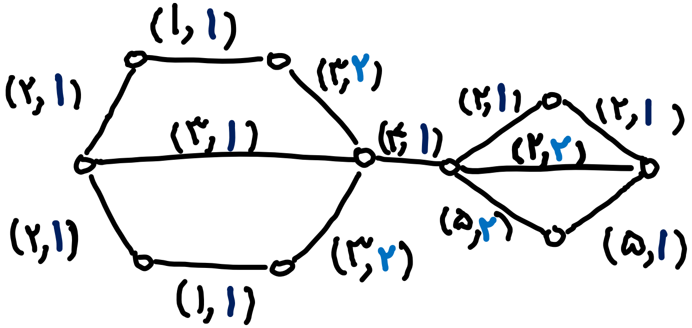


گراف : G



درخت MST : T
مورد نظر ما

تعیین اولویت لغت نامه‌ای برای وزن یال‌های G :



G

فرض کنید می‌خواهیم اجرای از کراسکال که T را به دست می‌آورد داشته باشیم.

$$w'(e) = (w(e), k)$$

$$e \in T \rightarrow k=1$$
$$e \notin T \rightarrow k=2$$

حالا ترتیب قاموسی را روی وزن یال‌ها در نظر بگیرید. هنگامی که به یال‌های هم وزن $w(e_1)=w(e_2)$ می‌رسیم اولویت با یال‌هایی است که $k=1$ باشد.

پس کراسکال، در نهایت درخت T را تولید می‌کند.

۹۴- فرض کنید در کدگذاری هافمن، طول کد همهٔ کاراکترها یکسان شده است. با فرض آن‌که تعداد کاراکترها ۳۲ می‌باشد، چند تا از گزاره‌های زیر همیشه درست است؟

- تعداد تکرار همهٔ کاراکترها یکسان است.
- اختلاف تکرار هر دو کاراکتر حداکثر یک است.
- اختلاف تکرار هر دو کاراکتر حداکثر دو است.
- به ازای هر عدد ثابت c می‌توان مثالی زد که دو کاراکتر وجود داشته باشند که اختلاف تکرارشان حداقل c باشد.

۱) ۰

۲) ۱

۳) ۲

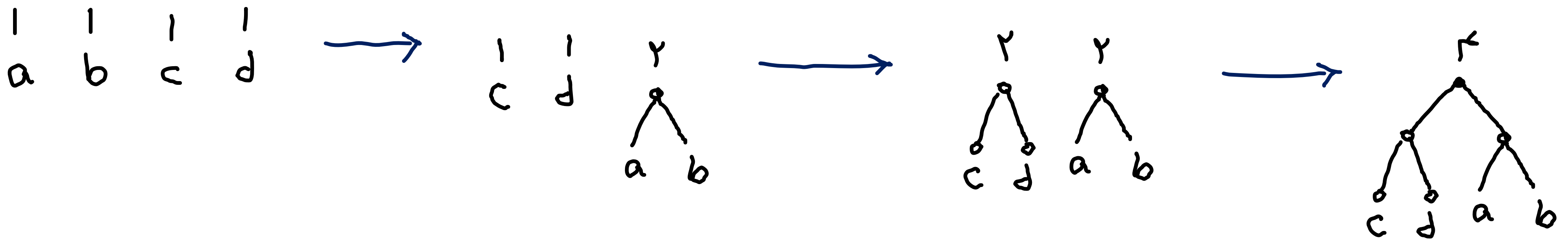
۴) ۳

پاسخ: گزینه (۲)

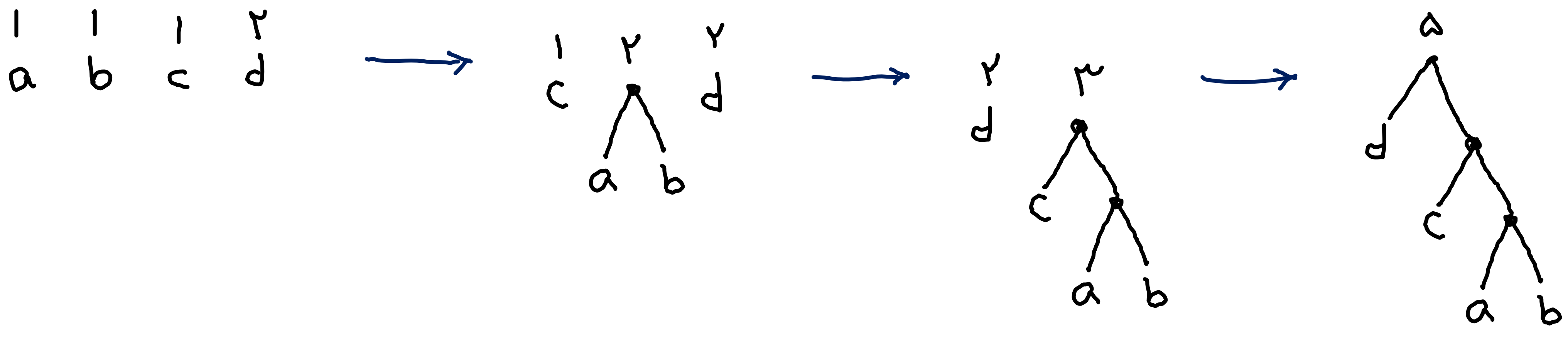
مسئله را به جای $n=2^5$ برای حالت $n=2^2$ بررسی می‌کنیم.
این کار را برای سادگی در بیان راه حل انجام می‌دهیم. شما می‌توانید همین
بررسی را برای $n=2^5$ تکرار کنید. در نهایت گزینه صحیح تغییری نمی‌کند.

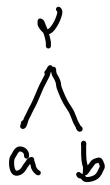
اگر تکرار همه کاراکترها برابر باشد، (مثلاً $f_i = 1$) می‌دانیم که طول کدها در حالت $n = 2^k$ برابر خواهد شد.

برای مثال برای $n = 4$:



اما در همین حالت $n=۴$ اگر به مثال زیر توجه کنید که حداکثر اختلاف فراوانی 1 است می بینیم طول کدها برابر نمی شود:



دقت کنید که در اولین مرحله، هنگامی که گره 

در صف اولویت درج می شود دلیلی ندارد که بعد از d درج شود.

۹۵- الگوریتم خرد کردن پول با روش حریصانهٔ «استفاده از پر ارزش‌ترین سکه، تا حد امکان» روی کدام مجموعهٔ سکه‌ها، لزوماً جواب بهینه (با کم‌ترین تعداد سکه) را تولید نمی‌کند؟ (فرض کنید از سکه‌های هر مجموعه به تعداد نامتناهی داریم.)

۱) $\{1,2,5\}$
۲) $\{1,4,7\}$
۳) $\{1,5,15\}$
۴) $\{1,7,15\}$

پاسخ: گزینه‌ی (۴).

یکی از مواردی که مورد توجه ما قرار می‌گیرد دنباله‌ی اختلاف‌های ارزش سکه‌ها در حالت صعودی است. ارزش سکه‌ها را به شکل صعودی مرتب کنید. پس به دنباله‌ی اختلاف‌ها توجه کنید. اگر این دنباله، نزولی (اکید) باشد علامت آن است که کاربرد روش حریصانه ما را به جواب بهینه می‌رساند. البته دقت کنید که این شرط، یک شرط لازم و کافی نیست. ما از این شرط به عنوان یک مسکن استفاده می‌کنیم.

$1 \xrightarrow{1} 2 \xrightarrow{3} 5$ $\quad$ $1 \xrightarrow{3} 4 \xrightarrow{3} 7$ $\quad$ $1 \xrightarrow{4} 5 \xrightarrow{5} 10$ $\quad$ $1 \xrightarrow{6} 7 \xrightarrow{3} 10$

در گزینه‌ی (۴) دنباله‌ی اختلاف‌ها نزولی اکید است. اگر بخواهیم $n = ۱۴$ دلار را با این سکه‌ها پرداخت، حالت بهینه آن است که ۲ سکه‌ی ۷ دلاری پرداخت کنیم. اما در روش حریصانه ابتدا یک سکه‌ی ۱۰ دلاری و سپس ۴ سکه‌ی یک دلاری پرداخت می‌کنیم که جواب بهینه نیست.

۹۶- فرض کنید در الگوریتم مرتب‌سازی سریع پس از عمل بخش‌بندی (Partition) آرایهٔ ⟨۳,۱,۲,۴,۵,۸,۷,۶,۹⟩ به‌دست آمده است. چند عدد از بین ۹ عدد در آرایه ممکن است محور این بخش‌بندی قرار گرفته باشند؟

۱) ۲   ۲) ۳   ۳) ۴   ۴) ۵

پاسخ: گزینه (۲)

بعد از اجرای عمل پارتیشن، اگر یه عنصر محور باشه، همه اعداد سمت چپ از آن کوچکتر مساوی هستند و همه اعداد سمت راست، از آن بزرگتر مساوی هستند. در این مثال، اعداد ۴ و ۵ و ۹ می‌توانند عنصر محور باشند.

۹۷- فرض کنید برای درهم‌سازی از روش زنجیره‌ای با یک جدول به اندازهٔ m استفاده شده است. تابع درهم‌ساز رکورد با کلید k را به خانهٔ $k \bmod m$ نگاشت می‌کند. اگر بدانیم کلید رکوردها، زیرمجموعهٔ $\left\{ i^{۲} \mid ۱ \leq i \leq ۱۰۰ \right\}$ است، به‌ازای کدام m، هزینهٔ جست‌وجو در بدترین حالت کم‌تر است؟

۱) ۷

۲) ۹

۳) ۱۱

۴) ۱۲

پاسخ: گزینه (۳)

با توجه به آن که $K=i^2$ ($1 \le i \le 100$) است ضابطه‌ی تابع درهم‌ساز به صورت $i^2 \bmod m$ است. هر چه تابع درهم‌ساز، به درهم‌سازی یکنواخت نزدیک‌تر باشد، حداکثر هزینه‌ی جستجو، کاهش می‌یابد.

می‌دانیم که در $i^2 \bmod m$ بهتر است m و ۲ نسبت به هم اول باشند پس $m=12$ مناسب نیست. همچنین بهترین انتخاب‌ها برای m عبارت است از اعداد اولی که از توان‌های ۲ تا حد امکان فاصله داشته باشند. به ویژه 2^a+1 و 2^a-1 انتخاب‌های خوبی نیستند پس 2^3+1 و 2^3-1 پیمانه مناسبی نیستند.

۹۸- کدام‌یک از دنباله‌های زیر (به ازای nهای بزرگ) بیش‌ترین ارتفاع ممکن را برای درخت هافمن ایجاد می‌کند؟ (اعضای دنباله‌ها نشان‌دهندهٔ تعداد تکرار کاراکترها در متن ورودی است نه خود کاراکترها.)

۱) دنبالهٔ از n عدد برابر

۲) دنبالهٔ از n عدد فیبوناچی پشت سر هم

۳) دنبالهٔ $(1,2,3,4,5,\ldots,n)$

۴) دنبالهٔ $(1^2,2^2,3^2,4^2,5^2,\ldots,n^2)$

پاسخ: گزینه (۲) [از ترفندات CLRS]

هرگاه دنباله‌ی $f_1, f_2, f_3, \ldots, f_n$ دنباله‌ی مرتب شده‌ی فراوانی‌ها باشد و داشته باشیم:

$$\sum_{i=1}^{m} f_i \le f_{m+2} \qquad (1 \le m \le n-2)$$

آنگاه ارتفاع درخت هافمن به بیشترین مقدار ممکن ($O(n)$) می‌رسد.

دو دنباله معروف که دارای این ویژگی هستند عبارتند از دنباله اعداد فیبوناتچی و دنباله هندسی $f_i = 2^i$.

(همچنین $f_i = A^i$ برای هر $A \ge 2$).

(*) بررسی سایر گزینه‌ها برای کامل کردن جواب :

در گزینه (۱) داریم $f_i = c$ و $f_{m+2} = c$:

$$\sum_{i=1}^{m} f_i = \sum_{i=1}^{m} c = mc > f_{m+2} = c$$

در گزینه (۳) داریم $f_i = i$ پس $f_{m+2} = m+2$:

$$\sum_{i=1}^{m} f_i = \sum_{i=1}^{m} i = \frac{m(m+1)}{2} > f_{m+2}$$

در گزینه (۴) داریم $f_i = i^2$ پس $f_{m+2} = (m+2)^2$:

$$\sum_{i=1}^{m} f_i = \sum_{i=1}^{m} i^2 = \frac{m(m+1)(2m+1)}{6} > f_{m+2}$$