به نام خداوند جان و خرد

آنچه تقدیم می شود، پاسخ تشریحی سئوالات درس ساختمان داده و طراحی الگوریتم در آزمون دکتری ۹۸ در گرایش نرم افزار است که سئوالات دکتری مهندسی فناوری اطلاعات ۹۸ نیز با آن اشتراک دارد.

با توجه به دست نویس بودن متن، نواقص احتمالی را با بزرگواری تحمل کنید

ابوالفضل تیک
مولف کتاب های داده الگوریتم، ریاضیات گسسته، هوش مصنوعی
انتشارات راهیان ارشد.

عضوی از مجموعه ی درخشان بابان

Scanned by CamScanner

کد کنترل

729

A

729A

تعداد سؤالات ساختمان داده و
طراحی الگوریتم : ۲۰ سوال

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

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

صبح جمعه
۹۷/۱۲/۳
دفترچهٔ شمارهٔ (۱)

آزمون ورودی دورهٔ دکتری (نیمه‌متمرکز) ـ سال ۱۳۹۸

رشتهٔ مهندسی کامپیوتر ـ نرم افزار و الگوریتم
کد (۲۳۵۴)

تعداد سؤال: ۴۵

مدت پاسخ‌گویی: ۱۵۰ دقیقه

عنوان مواد امتحانی، تعداد و شمارهٔ سؤالات

ردیف	مواد امتحانی	تعداد سؤال	از شمارهٔ	تا شمارهٔ
۱	مجموعه دروس تخصصی: ساختمان داده‌ها و طراحی الگوریتم‌ها ـ سیستم‌های عامل پیشرفته ـ پایگاه داده‌های پیشرفته	۴۵	۱	۴۵

این آزمون نمرهٔ منفی دارد.

استفاده از ماشین حساب مجاز نیست.

۱۳۹۸

* داوطلب گرامی، عدم درج مشخصات و امضا در مندرجات جدول ذیل، به‌منزلهٔ عدم حضور شما در جلسهٔ آزمون است.

اینجانب .... با شمارهٔ داوطلبی .... در جلسهٔ این آزمون شرکت می‌نمایم. امضا:

۱- یک ماتریس دو بعدی $n \times n$ از اعداد داده شده، که اعداد هر سطر و هر ستون آن مرتب شده است. به ازای عدد دادهٔ شدهٔ x، جست و جوی x در این ماتریس در چه زمانی امکان‌پذیر است؟

۱) $O(n)$

۲) $O(\log n)$

۳) $O(\log^2 n)$

۴) $O(n \log n)$

۲- می‌خواهیم بزرگ‌ترین زیر دنبالهٔ مشترک دو دنبالهٔ $a_1,\dots,a_n$ و $b_1,\dots,b_m$ را محاسبه کنیم. فرض کنید $L(i,j)$ برابر طول بزرگترین زیر دنبالهٔ مشترک $a_1,\dots,a_i$ و $b_1,\dots,b_j$ باشد. کدام‌یک از تعاریف بازگشتی زیر درست است؟

الف) $L(n,m) = \max(L(n-1,m), L(n,m-1), L(n-1,m-1)+1 \text{ if } a_n = b_m)$

ب) $L(n,m) = \max(L(n-1,m), L(n-1,k-1)+1)$ که k برابر بزرگ‌ترین عددی است که $a_n = b_k$ (درصورت عدم وجود $k = 0$ خواهد بود)

فرض کنید $L(i,0) = L(0,i) = 0$ و $L(i,-1) = -1$ برای هر $i \geq 0$.

۱) فقط الف

۲) فقط ب

۳) الف و ب

۴) هیچ‌یک از الف و ب

۳- فرض کنید یک آرایهٔ دوبعدی $m \times n$ در اختیار داریم که هر ردیف آن مرتب شده است. فرض کنید همهٔ اعداد متمایز هستند. می‌خواهیم kاُمین عدد در آرایه را پیدا کنیم. در چه زمانی این کار امکان‌پذیر است؟

۱) $O(m.n)$

۲) $O(\log n \log m)$

۳) $O(\log n + \log m)$

۴) $O(m(\log n + \log m))$

۴- اگر ظرفیت همهٔ یال‌ها در یک شبکه برابر C باشد، زمان اجرای الگوریتم فورد ـ فالکرسون برای محاسبه شار بیشینه از مبدا s به مقصد t در بدترین حالت کدام مورد خواهد بود؟

(فرض کنید تعداد رئوس و یال‌های گراف به ترتیب n و m هستند و درجه خروجی s برابر k باشد. همچنین فرض کنید در هر مرحله الگوریتم بیش‌ترین شار ممکن را از مسیر انتخاب شده، عبور می‌دهد.)

۱) $O(kC + m + n)$

۲) $O(kC(m+n))$

۳) $O(C(m+n))$

۴) $O(k(m+n))$

۵- فرض کنید ۱۳۹۷ نقطه متمایز روی محور اعداد حقیقی داده شده است. می‌خواهیم این ۱۳۹۷ نقطه را طوری رنگ‌آمیزی کنیم که به ازای هر بازهٔ $[a,b]$ روی محور اعداد حقیقی، از بین نقاطی که در این بازه قرار گرفته‌اند حداقل یک نقطه وجود داشته باشد که رنگ آن با بقیه نقاط داخل بازه متفاوت باشد. حداقل چند رنگ برای این کار نیاز است؟

۱) ۶

۲) ۱۱

۳) ۳۸

۴) ۱۳۹۷

۶- فرض کنید یک B-tree داریم با n برگ که درجهٔ هر گره حداقل $\log n$ و حداکثر $2\log n - 1$ است. هزینهٔ جست‌وجوی یک عدد در این درخت کدام است؟ (فرض کنید کلیدها داخل هر گره میانی در یک لیست پیوندی یک سویه ذخیره شده‌اند.)

۱) $O(\log n)$
۲) $O(\log n \log\log n)$
۳) $O(\log n \log^2 \log n)$
۴) $O(\log^2 n / \log\log n)$

۷- فرض کنید یک گراف وزن‌دار همبند داده شده است که وزن یال‌ها متمایز است. یک یال را امن گوییم اگر در هیچ دوری حضور نداشته باشد و یک یال را خطرناک گوییم اگر سنگین‌ترین یال در یک دور باشد. کدام‌یک از دو گزاره زیر درست است؟

الف) هر یال امن عضو درخت پوشای کمینه است.

ب) هر یال خطرناک عضو درخت پوشای کمینه نیست.

۱) الف
۲) ب
۳) الف و ب
۴) هیچ‌یک از الف و ب

۸- گراف جهت‌دار G با n رأس و m یال داده شده است. هر رأس i از گراف ارزشی به اندازهٔ V_i دارد. به‌ازای هر رأس i از گراف، با ارزش‌ترین رأسی که از رأس i قابل دسترسی است را W_i می‌نامیم. می‌خواهیم تمام W_i ها را به‌ازای i از ۱ تا n محاسبه کنیم. این کار در چه زمانی قابل انجام است؟ (بهترین گزینه را انتخاب کنید.)

۱) $O(m+n)$
۲) $O(m+n^2)$
۳) $O(n(m+n))$
۴) $O(m+n\log n)$

۹- یک درخت جستوجوی دودویی با n گره داریم که به علت نویز، اعداد ذخیره شده در برخی از گره‌های آن تغییر کرده است. تنها عملی که می‌توان برای اصلاح این درخت انجام داد جابه‌جا کردن مقادیر ذخیره شده در یک گره و یکی از فرزندان آن است. کمینهٔ تعداد اعمال مورد نیاز برای تبدیل درخت به یک درخت دودویی جست‌وجو در بدترین حالت کدام است؟ (دقت کنید که درخت اولیه لزوماً متوازن نیست.)

۱) $O(n)$
۲) $O(n^2)$
۳) $O(n\log n)$
۴) $O(n\log\log n)$

۱۰- زوج‌های مرتب زیر را در نظر بگیرید:

(۱۰ , A) , (۲, B), (۵, C), (۷, D), (۸, E), (۱, F), (۴, G)

فرض کنید درختی داریم که براساس مؤلفه‌های اول این زوج‌ها یک هرم کمینه، و براساس مؤلفه‌های دوم یک درخت جست‌وجوی دودویی است. ارتفاع این درخت کدام است؟

۱) ۲

۲) ۳

۳) ۴

۴) ۵

۱۱- در گراف همبند و بدون جهت G با n رأس، از یک رأس مشخص BFS و DFS را اجرا می‌کنیم. ترتیب ملاقات رئوس در هر دو اجرا یکسان شده است. در این خصوص کدام مورد درست است؟

۱) گراف G فقط ستاره‌ای است.

۲) گراف G فقط یک مسیر است.

۳) تعداد یال‌های G از $O(n)$ است.

۴) تعداد یال‌های G می‌تواند $\Omega(n \log n)$ باشد.

۱۲- فرض کنید گراف G همبند، بدون جهت و وزن‌دار است به‌طوری‌که می‌تواند دور منفی هم داشته باشد. در مورد مسئله پیدا کردن کوتاه‌ترین مسیر از یک رأس به رأس دیگر طوری که از هر رأسی حداکثر یکبار عبور کند، چه می‌توان گفت؟

۱) یک مسئله ان‌پی ـ تمام است.

۲) یک مسئله ان‌پی ـ سخت است.

۳) به علت وجود دور منفی در گراف لزوماً چنین مسیری وجود ندارد.

۴) در زمان چندجمله‌ای برحسب اندازهٔ ورودی می‌توان مسئله را حل کرد.

۱۳- تعدادی فایل با اندازه‌های مشخص را می‌خواهیم روی نوار ذخیره کنیم. فرض کنید $f_1,...,f_n$ به‌ترتیب (از راست به چپ) روی نوار ذخیره شده باشند. هزینه خواندن فایل f_i برابر $\sum_{k=1}^{i} |f_k|$ خواهد بود که $|f_k|$ برابر طول فایل f_k می‌باشد. فرض کنید قرار است هر فایل تنها یکبار خوانده شود. می‌خواهیم مجموع هزینه را کمینه کنیم. بدین منظور از الگوریتم حریصانه زیر استفاده می‌کنیم. فایل‌ها را به‌ترتیب اندازه از کوچک به بزرگ روی نوار ذخیره می‌کنیم. اگر n تعداد فایل‌ها باشد، کم‌ترین n که به ازای آن الگوریتم فوق لزوماً درست کار <u>نمی‌کند</u>، کدام است؟

۱) ۲

۲) ۳

۳) ۴

۴) به‌ازای هر n، الگوریتم فوق بهینه عمل می‌کند.

۱۴- آرایهٔ A شامل n عنصر داده شده است. می‌دانیم که تمام عناصر به‌جز $\sqrt[4]{n}$ عنصر، در محل مرتب‌شدهٔ خود هستند ولی مکان عناصر نامرتب را نمی‌دانیم. این آرایه را در چه زمانی می‌توان مرتب کرد؟

۱) $O(n)$ ۲) $O(n\sqrt[4]{n})$

۳) $O(n \log n)$ ۴) $O(\sqrt{n} \log n)$

۱۵- پیمایش‌های پیش ترتیب و پس‌ترتیب یک درخت دودویی به‌صورت زیر است:

preorder: abedefg , postorder : cbfgeda

با فرض ذخیره‌سازی درخت در آرایه (ریشه در خانه‌ی ۱ و فرزندان گره اندیس در اندیس‌های $2i$ و $2i+1$)، حداکثر تعداد خانه‌های بلا استفاده قبل از محل آخرین گره در آرایه کدام است؟

۱) ۵

۲) ۶

۳) ۷

۴) ۸

۱۶- یک ساختمان داده را درنظر بگیرید که از دو پشته S_1 و S_2 تشکیل شده است. این ساختمان داده دو عمل درج و استخراج را پشتیبانی می‌کند. به هنگام درج عنصر x در این ساختمان داده، $push(S_1, x)$ را اجرا می‌کنیم. به هنگام استخراج اگر S_2 خالی نبود، $Pop(S_2)$ را اجرا می‌کنیم. در غیر این‌صورت همهٔ عناصر داخل S_1 را پاپ و داخل S_2 پوش می‌کنیم و بعد دستور $Pop(S_2)$ را اجرا و به عنوان خروجی دستور استخراج در نظر می‌گیریم. اگر دو پشته در ابتدا خالی باشد و n عمل درج و استخراج به‌ترتیب دلخواه انجام شود، هزینه سرشکن این عمل‌ها کدام است و ساختمان داده فوق چه ساختمان داده‌ای را پیاده‌سازی می‌کند؟

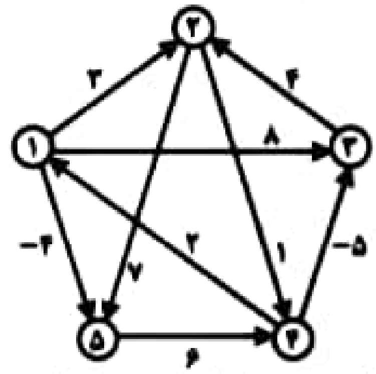
۱) $O(1)$ و صف

۲) $O(n)$ و صف

۳) $O(1)$ و پشته

۴) $O(n)$ و پشته

۱۷- اگر الگوریتم جانسون برای یافتن کوتاه‌ترین مسیر بین تمام رأس‌های گراف را روی گراف وزن‌دار زیر اجرا کنیم، پس از اجرای مرحلهٔ تغییر وزن یال‌ها در الگوریتم، وزن جدید یال بین رأس‌های ۱ و ۵ که وزن اولیهٔ آن ۴- است، کدام مقدار خواهد شد؟



۱) ۴-

۲) ۴

۳) ۳

۴) ۰

۱۸- اگر رشتهٔ **ababbcbaabdadad** را به‌وسیلهٔ الگوریتم هافمن کدگذاری کنیم، طول رشتهٔ حاصل چند بیت خواهد بود؟

۱) ۲۰

۲) ۲۴

۳) ۲۸

۴) ۳۰

۱۹- فرض کنید n عدد صحیح k بیتی داریم. فرض کنید هزینه جمع، تفریق و مقایسه دو عدد k بیتی $O(k)$ است. اگر $k = O(\log n)$ باشد، کدام گزینه در مورد الگوریتم‌های مرتب‌سازی درست است؟

۱) زمان اجرای الگوریتم مرتب‌سازی شمارشی $O(n)$ است.

۲) زمان اجرای الگوریتم مرتب‌سازی سریع $O(n \log n)$ است.

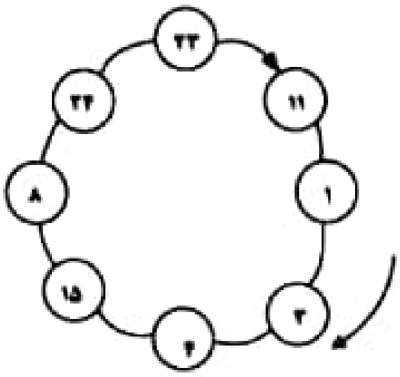
۳) زمان اجرای الگوریتم مرتب‌سازی ادغامی $O(n \log^2 n)$ است.

۴) زمان اجرای الگوریتم مرتب‌سازی درجی $\Omega(n^2 \log^2 n)$ است.

۲۰- آرایه‌ای شامل n عدد داریم. اگر در اجرای الگوریتم مرتب‌سازی ادغامی روی این آرایه هرگاه تعداد اعداد کمتر از $\sqrt{n}$ شد، روال بازگشتی را متوقف و از الگوریتم مرتب‌سازی درجی استفاده کنیم، زمان اجرای الگوریتم کدام مورد خواهد بود؟ (فرض کنید زمان اجرای الگوریتم مرتب‌سازی درجی از مرتبهٔ $O(m^2)$ است که m تعداد اعداد می‌باشد.)

۱) $O(n^2)$
۲) $O(n\sqrt{n})$
۳) $O(n\log n)$
۴) $O(n\log n\sqrt{n})$

۲۱- انتخاب حلقوی رهبر (ring-based leader election) (الگوریتم chang & Robert) برای ring زیر استفاده شده است. فرض کنید که در حال حاضر رهبری وجود ندارد و خطایی نیز رخ نمی‌دهد. در بهترین حالت کدام پردازه می‌بایست انتخاب رهبر را آغاز نماید؟



۱) ۱
۲) ۱۵
۳) ۲۳
۴) ۲۴

۲۲- کدام مورد از مزایای یک سیستم توزیع شده نیست؟

۱) Reliability
۲) Incremental growth
۳) Resource sharing
۴) هیچ‌کدام

۲۳- در پیاده‌سازی الگوریتم‌های mutual exclusion در یک سیستم توزیع شده با n گره، توسط الگوریتم Ricart ـ Agrawala، چند پیام در هر مرحله فرستاده می‌شود؟

۱) $2n$
۲) $2n-1$
۳) $2(n-1)$
۴) $2(n+1)$

۲۴- در یک سیستم توزیع شده، پردازه‌ها به‌صورت همروند اجرا می‌شوند و به وسیله پیام با هم ارتباط برقرار می‌کنند. کدام مورد در پیام استفاده نمی‌شود؟

۱) نام فرستنده
۲) زمان دریافت
۳) نام دریافت کننده
۴) زمان مجازی ارسال

۲۵- فرض کنید که ۴ پردازه با Causal ordering داریم که بردار فعلی آن‌ها در جدول زیر داده شده است.

پردازه	بردار
A	(۳،۵،۲،۱)
B	(۲،۵،۲،۱)
C	(۳،۵،۲،۱)
D	(۳،۴،۲،۱)

اگر A یک پیام ارسال نماید، کدام پردازه‌ها بلافاصله می‌توانند آن را دریافت کنند؟

۱) A و C
۲) A و D
۳) B و C
۴) B و D

۱

گزینه (۱)

حداکثر به n حرکت عمودی و n حرکت افقی نیاز داریم: $۲n = O(n)$.

۱٫۵	۲٫۲	۳٫۷	۴	۴٫۵	۵
۲	۳٫۱	۶٫۵	۸	۹٫۱	۱۰
۳٫۲	۴	۷٫۲	۸٫۱	۱۰	۱۱٫۵
۴٫۱	۷٫۵	۷٫۶	۹	۱۱	۱۲٫۵
۶٫۲	۸	۸٫۵	۱۰	۱۱٫۱	۱۳
۷	۹	۹٫۱	۱۲	۱۳٫۲	۱۵

مسیر جستجوی $x = ۷٫۵$

ابتدا x را با درایه های ستون آخر مقایسه می کنیم. [به عنوان مثال جستجوی عدد $x = ۷٫۵$ را در شکل می بینید. $x > ۵$ پس به ستون بعدی می رویم.
$x < ۱۰$ پس حرکت ستونی را به سطری تبدیل می کنیم. تا جایی که x کوچکتر از اعداد باشد ادامه می دهیم: $x < ۹٫۱$ ، $x < ۸$ ، اما $x > ۶٫۵$
پس حالا ستونی حرکت می کنیم: $x > ۷٫۲$ ، اما $x < ۷٫۶$ پس ادامه ی حرکت سطری می شود و در نهایت $x = ۷٫۵$ یافت می شود.

خلاصه: در ابتدا از $i=1$ ، $j=n$ آغاز می‌کنیم

اگر $x > a_{ij}$ آنگاه $j=j+1$

اگر $x < a_{ij}$ آنگاه $i=i-1$.

اگر هم $j>n$ یا $i<1$ شد، جستجو ناموفق بوده است.

۲ گزارش (۳)

فرمول (الف) همان شیوه‌ی معروف محاسبه‌ی LCS است.
اگر $a_n = b_m$ باشه، یک کاراکتر مشترک یافته‌ایم و حالا باید n-1 مولفه‌ی A را با m-1 مولفه‌ی B مقایسه کنیم:

$$\text{if } a_n = b_m \Rightarrow L(n,m) = 1 + L(n-1, m-1)$$

اگر $a_n \neq b_m$ باشه باید تمام مولفه‌های A را با m-1 مولفه‌ی B و همچنین تمام مولفه‌های B را با n-1 مولفه‌ی A در نظر بگیریم و هر کدام LCS بزرگتری داشتند، جواب هستند.

در این حالت داریم:

$$L(n,m) = \max\left(L(n-1,m) \text{،} L(n, m-1) \right)$$

حالا اگر بخواهیم همه‌ی موارد بالا را در یک فرمول خلاصه کنیم، فرمول (الف) حاصل می‌شود.

برای اثبات فرمول (ب) فرض کنیم K بزرگترین اندیس با ویژگی $a_n = b_K$ باشد. اکنون دو حالت داریم:

۱) a_n در بزرگترین زیررشته‌ی مشترک، نقش داشته باشد:

$$b_1\ b_2\ \dots\ b_k\ \ b_{k+1} \dots b_m$$

$$a_1\ a_2\ \dots\dots\ a_n$$

از آنجا که مولفه‌های b_j $(j>k)$ با a_n برابر نخواهند بود

خواهیم داشت:

$$L(n,m) = 1 + L(n-1, k-1)$$

۲) حالت دوم آن است که a_n در بزرگترین زیررشته‌ی مشترک دخالتی نداشته باشد در این صورت باید $n-1$ مولفه‌ی A را با تمام مولفه‌های B در نظر بگیریم:

$$L(n,m) = L(n-1, m)$$

بدیهی است که از بین حالات (۱) و (۲) هر کدام بزرگتر باشد جواب LCS است. پس (ب) صحیح است.

۳ گزینه (۴)

یک مسأله‌ی معروف است. فرض کنیم لیست‌های $A_1, \dots, A_m$ هر کدام مرتب شده باشند. (همگی صعودی یا همگی نزولی) منظور از مرتب در سؤالات، مرتب صعودی (غیرنزولی) است.

در ضمن فرض کنید مجموع کل اعداد یعنی تعداد کل آنها N باشد.

در این صورت محاسبه‌ی میانه و همچنین محاسبه‌ی آماره‌ی ترتیبی (k امین کوچکترین عدد) در کل این اعداد از مرتبه‌ی زمانی زیر است:

$$O(\underbrace{m}_{\text{تعداد لیست‌ها}} \log \underbrace{N}_{\text{تعداد کل اعداد}})$$

در این مثال ما هر سطر ماتریس را یک لیست مرتب فرض کنیم.

پس m لیست مرتب داریم و در هر کدام n عدد داریم یعنی $N = mn$ است.

$$\text{جواب} = O(m \log mn)$$
$$= O(m(\log m + \log n))$$

۴ گزینه (۴)

از آنجاکه درجه‌ی خروجی s برابر K است و

ظرفیت همه‌ی یال‌ها برابر است و هر بار از بیشترین ظرفیت

ممکن برای هر مسیر استفاده می‌کنیم خواهیم داشت:

اولاً: برای یافتن مسیری از s به t از BFS

استفاده می‌کنیم که مرتبه‌ی $O(V+E)$ دارد.

ثانیاً، تعداد یال‌های این مسیر که باید ظرفیت جدید آنها

محاسبه شود حداکثر E است. پس تا اینجا به مرتبه‌ی

$O(V+2E)$ می‌رسیم.

ثالثاً: حداکثر K مسیر از s به t خواهیم یافت پس

مراحل قبلی حداکثر K بار تکرار می‌شوند.

$$O(K(V+2E)) = O(K(n+2m))$$

با حذف ضریب ثابت ۲ به گزینه (۴) می‌رسیم.

[می‌دانیم که $2m = \Theta(m)$]

توجه مهم: دقت کنید که چون ظرفیت همه یال‌ها برابر با C است

لازم نبود min ظرفیت مسیر را محاسبه کنیم.

۵ گزینه (۲)

برای رسیدن به مقصود یک رنگ با شماره‌ی (۱) را به نقطه‌ی وسط خواهیم داد.
یعنی نقطه ای که تعداد نقاط چپ و راست آن $\lfloor \frac{n-1}{2} \rfloor$ و $\lceil \frac{n-1}{2} \rceil$ باشد.
از این رنگ دیگر استفاده نخواهیم کرد.
زیرا می‌خواهیم اگر همه‌ی نقاط داخل بازه‌ی بزرگ $[a, b]$ افتادند، یک
نقطه با رنگ منحصر به فرد داشته باشیم.
اکنون اگر مساله را برای یک نیمه، حل کنیم می‌توان دقیقاً همان رنگ‌ها
را برای نیمه‌ی دیگر هم استفاده کرد. پس داریم:

$$\begin{cases} T(n) = T\left(\frac{n-1}{2}\right) + 1 \\ T(1) = 1 \end{cases}$$

می دانیم که پاسخ این رابطه ی بازگشتی تقریباً $\lceil \log n \rceil$ است. پس به ازای $n = 1397$ داریم

$$\lceil \log 1397 \rceil \simeq 11$$

دقت کنید که $2^{10} = 1024$ و $2^{11} = 2048$.

البته می توانید برای اطمینان بیشتر $T(1397)$ را محاسبه کنید:

$$T(1397) = T(698) + 1$$

$$T(698) = T(348) + 1$$

$$\vdots$$

$$T(5) = T(2) + 1$$

$$T(2) = T(1) + 1$$

حالا $T(1) = 1$ است و $T(1397) = 11$ حاصل می شود.

۶ـ گزینه (۴) [کلید اولیه نادرست است]

هزینه‌ی جستجو در یک B.tree با حداکثر ارتفاع h که در هر گره آن t کلید در یک لیست پیوندی یک سویه قرار داشته باشد از مرتبه‌ی $O(th)$ است. در واقع $O(h)$ برای یافتن گره و سپس $O(t)$ برای جستجو در کلیدهای آن گره.

در این مثال، تعداد فرزندان هر گره میانی $\Theta(\log n)$ است پس تعداد کلیدهای درج شده در هر گره $t = \Theta(\log n)$ است. [می‌دانیم که تعداد فرزندان را $t+1$ نشان می‌دهد]

برای یافتن حداکثر ارتفاع باید از حداقل ظرفیت‌ها استفاده کنیم پس فرض کنیم درجه‌ی گره‌های میانی $\log n$ باشد. در عمق صفر 1 گره داریم. در عمق بعدی $\log n$ گره، در عمق ۲ تعداد گره‌ها $\log^2 n$ است و به همین ترتیب در عمق h تعداد گره‌ها $(\log n)^h$ است.

طبق صورت سؤال تعداد برگ ها باید n باشد پس: $n = (\log n)^h$

از طرفین log می گیریم:

$$\log n = h \log\log n$$

$$\Rightarrow h = \frac{\log n}{\log\log n}$$

$$\text{جواب} = O(t.h) = O\left(\frac{\log^2 n}{\log\log n}\right)$$

۷

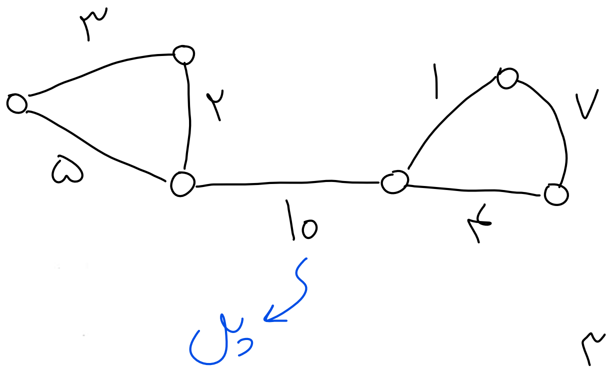
گزینه (۳)

جزوه‌ی MST را از کانال‌های بابان مطالعه کنید.

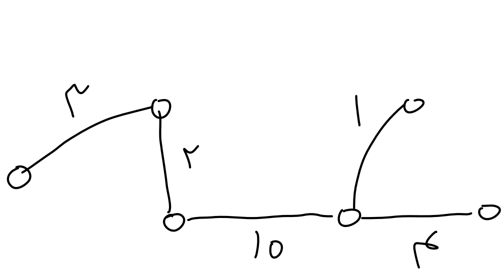
یالی که در هیچ دوری شرکت ندارد یک یال برشی (پل) است پس حتماً در MST قرار دارد.

در ضمن سنگین‌ترین یال هر دور، قطعاً عضو MST نیست. دقت کنید که وزن یال‌ها متمایز است و در نتیجه هر دور فقط یک یال با عنوان سنگین‌ترین یال دارد.

G:



MST:



۸ گزینه (۱)

گراف $G = (V, E)$ را داریم. ابتدا جهت یال‌ها را معکوس می‌کنیم. پس هر یال $(i, j) \in E$ به یال $(j, i) \in E'$ تبدیل می‌شود. حالا گراف G' را داریم که جهت یال‌هایش عکس G است. مرتبه‌ی زمانی این کار: $O(m)$.

بزرگترین v_i را به دست آوریم: $O(n)$. فرض کنید i_0 بیشترین مقدار v_{i_0} را دارد. $DFS(i_0)$ را اجرا کنید. با ملاقات هر گره جدید مانند i توقف کنید $W_i = i_0$. (اجرای DFS از مرتبه‌ی $O(m+n)$ است.)

از آنجا که DFS برای سایر مولفه‌ها نیز ادامه می‌یابد، مرتبه‌ی زمانی در مجموع برابر است با:

$$O(m+n + n+m) = O(m+n)$$

۹ گزینه (۲)

یک مساله‌ی معروف و بسیار پرتکرار است. پاسخ آن در حالت کلی $O(m$ [illegible]$)$ است که [illegible] تعداد گره‌هایی است که کلید ذخیره شده در آن دچار تغییر شده است و h ارتفاع درخت است.

در این مساله، ما نمی‌دانیم چه تعدادی از گره‌ها دچار تغییر شده‌اند پس حداکثر m را در نظر می‌گیریم یعنی $m \leq n$.

در ضمن نمی‌دانیم که درخت BST متوازن بوده یا خیر پس حداکثر ارتفاع را در نظر می‌گیریم یعنی $h \leq n$.

$$\text{پاسخ} = O(m \times h) = O(n \times n) = O(n^2)$$

توجه: اگر درخت اولیه، متوازن بود جواب $O(n \log n)$ می‌شد. مثلاً اگر AVL دچار نوبنیان شود پاسخ مساله $O(n \log n)$ است.

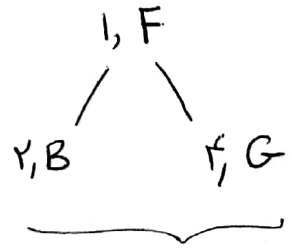
۱۵ گزینه (۳) [فیلم آموزشی Treap را در کانال دی بابان مشاهده کنید]

مولفه‌ی اول که هرم کمینه را می‌سازند، به ما اولویت درج را می‌دهند:

(1, F) , (2, B) , (4, G) , (5, C) , (7, D) , (8, E) , (10, A)

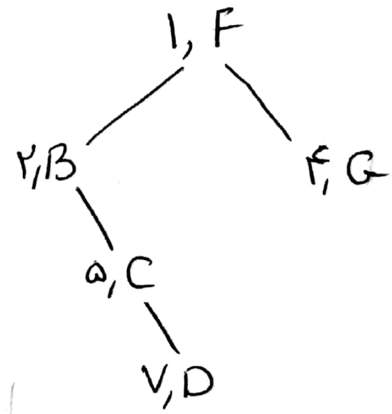
$\xrightarrow{\hspace{2em}}$
اولویت درج

حالا این‌ها را به ترتیب و با توجه به ترتیب حروف الفبا
یعنی مولفه‌ی دوم، در درخت BST درج کنیم.



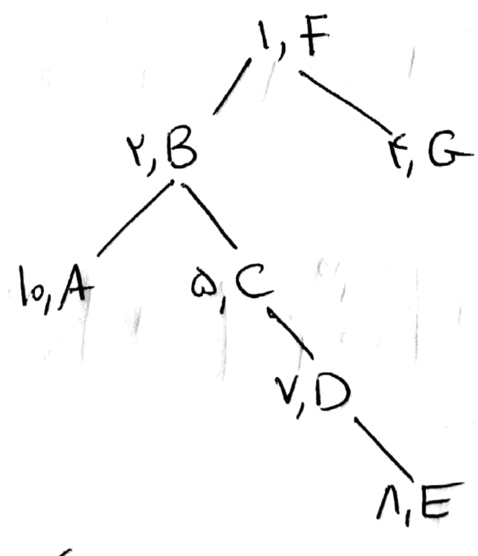
درج ۳ تای اول

$\Longrightarrow$



درج C
و سپس D

$\Longrightarrow$

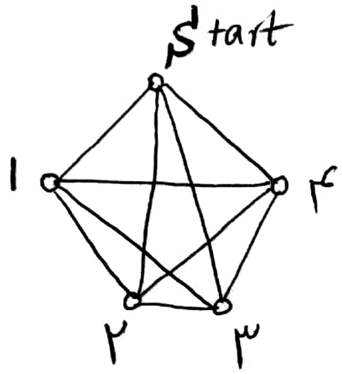


در نهایت این درخت
به دست می‌آید:
ارتفاع آن $h=4$ است.

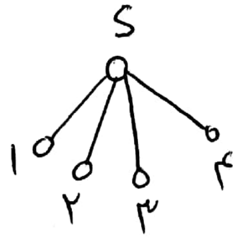
توجه: تعداد Treap ها منحصر به فرد است یعنی فقط یک Treap
با این کلیدها وجود دارد.

۱۱ گزینه (۴) [کلید اولیه گزینه (۳) بوده است]

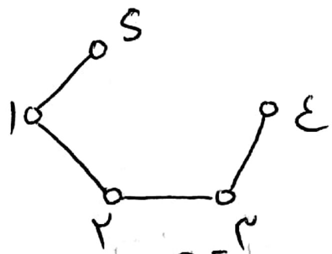
حتی در یک گراف کامل که تعداد یال‌ها بیش $E = \frac{n(n-1)}{2} = \Theta(n^2)$ است،
می‌توان اولویت گره‌ها را طوری تعریف کرد که ترتیب ملاقات گره‌ها در
DFS و BFS یکسان شود. مثلاً:



با شروع از گره s در گراف K_5 طوری که
اولویت‌ها مشخص شده،



BFS(s)



DFS(s)

هر دو الگوریتم BFS و DFS به ترتیب s، ۱، ۲، ۳، ۴ را ملاقات می‌کنند.

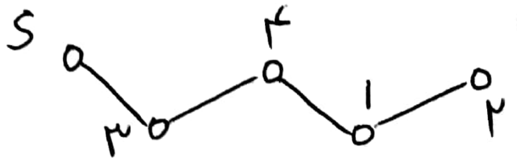
بنابراین با این استدلال گزینه (۴) صحیح است.

" طراح سؤال نتوانسته منظورش را به خوبی بیان کند."

چند نکته :

در این زمینه بهتر است موارد زیر را هم بدانید :

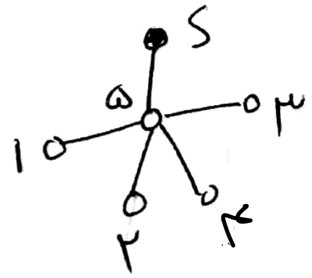
۱) مسیر ساده‌ای که با گره S آغاز می‌شود،



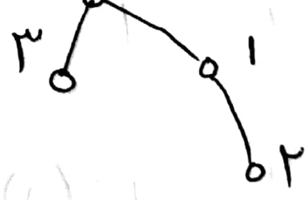
گرافی است با این ویژگی که ترتیب ملاقات گره‌ها در BFS(S) و DFS(S) بدون توجه به آن که چه اولویت‌هایی برای گره‌ها تعریف کرده باشید، همواره یکسان است.

در این شکل هر دو الگوریتم به ناچار به ترتیب S، ۳، ۴، ۱، ۲ را ملاقات می‌کنند. البته اگر از S جستجو را آغاز کنید.

۲) گراف ستاره‌ای هم ویژگی بالا را دارد. در این گراف، مهم نیست که از کدام گره BFS و DFS را آغاز کنید. ترتیب ملاقات‌ها یکسان خواهد بود.



۳) همه‌ی درخت‌ها چنین نیستند مثلاً،

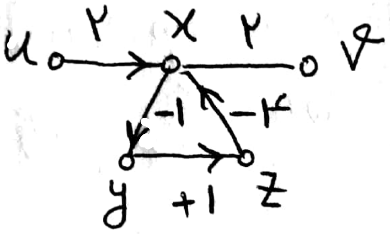


در BFS : S، ۱، ۳، ۲

در DFS : S، ۱، ۲، ۳

۱۲ گزینه (۲)

وقتی در گراف وزن دار، دوری با وزن منفی وجود داشته باشد، مسائلی کوتاه‌ترین مسیر که بین برخی از جفت گره های $(u،v)$ ممکن است جواب نداشته باشد. مثلاً در این شکل می توانیم:



$u \ \underbrace{xyz\ \ xyz\ \ xyz\ \ldots\ldots}_{\text{تعداد دلخواه}}\ x\ v$

دور با وزن منفی را هر چند بار که بخواهیم طی کنیم پس اصولاً کوتاه‌ترین (کم هزینه ترین) مسیر از u به v وجود ندارد.

اما اگر مانند صورت سؤال شرط بگذاریم که در مسیرهای مورد نظر از هر راس حداکثر یک بار عبور می کنیم، حالا دیگر دورهای موجود، قابل استفاده نیستند اما باز هم ما الگوریتم های معمول مانند فلوید، جانسون، ... را از دست می دهیم.

این مسأله، مشابه مسأله‌ی فروشنده‌ی دوره‌گرد است. در آنجا ما می‌خواهیم

از هر راس دقیقاً یک بار عبور کنیم و به جای اول برگردیم. در این مسأله می

خواهیم از u به v برویم و از هر راس حداکثر یک بار گذر کنیم. بنابراین

چون فروشنده‌ی دوره‌گرد NP-Hard است این مسأله هم باید از نوع

NP-Hard باشد. [الگوریتمی که فروشنده‌ی دوره‌گرد را حل کند، این مسأله

را هم حل کرده است]

Scanned by CamScanner

۱۳ گزینه (۴) (دقت کنید صورت سؤال $\sum_{k=1}^{n} |f_k|$ است)

برای توضیح راحت‌تر، فرض کنید ۵ فایل داریم و به این صورت روی نوار ذخیره شده باشند:

f_1	f_2	f_3	f_4	f_5

هزینه‌ی خواندن f_1 : $|f_1|$

هزینه‌ی خواندن f_2 : $|f_1|+|f_2|$

" " f_3 : $|f_1|+|f_2|+|f_3|$

" " f_4 : $|f_1|+\cdots+|f_4|$

هزینه‌ی " f_5 : $|f_1|+\cdots+|f_5|$

بنابراین:

$$T(n) = 5|f_1| + 4|f_2| + 3|f_3| + 2|f_4| + |f_5|$$

پس مثل ایده‌ای که به کار رفته در کد هافمن، بهتر است $|f_1|$ که ضریب بزرگ‌تری دارد، فایلی با کمترین اندازه باشد و $|f_5|$ که ضریب کوچک‌تری دارد فایل با اندازه‌ی بزرگ‌تر باشد.

این عملیات در ۳ مرحله قابل انجام است. در مرحله اول ما عناصر مطلوب (مرتب) و نامطلوب را شناسایی کرده و آنها را در ۲ لیست A_1 و B_1 قرار می‌دهیم. A_1 یک لیست مرتب شامل $n-\sqrt[4]{n}$ عضو است و B_1 لیست به طول $\sqrt[4]{n}$ و احتمالاً نامرتب است. [این گام از مرتبه‌ی $O(n)$ است و در پایان نحوه‌ی انجام آن را توضیح می‌دهیم]

مرحله‌ی دوم، مرتب کردن لیست B_1 است. از آنجا که اندازه‌ی آن $\sqrt[4]{n}$ است پس حداکثر نیاز به $O\left(\sqrt[4]{n}^{\,2}\right) = O(\sqrt{n})$ مقایسه دارد.

مرحله سوم: ادغام لیست‌های مرتب شده که حداکثر نیاز به این تعداد مقایسه دارد: $O(n-\sqrt[4]{n} + \sqrt[4]{n}) = O(n)$

بنابراین در مجموع این ۳ مرحله داریم:

$$T(n) = O(n + \sqrt{n} + n) = O(n)$$

<u>توضیح مرحله‌ی اول:</u> مشابه ثبت رکورد (روال استخدام) است.

ابتدا $RECORD = A[1]$ است. برای $i = 2, \dots, n$

اگر $A[i]$ از رکورد بزرگتر مساوی باشد، آن را به یکی از لیست‌ها می‌بریم و درضمن $RECORD = A[i]$ می‌شود.

اگر $A[i]$ از رکورد کمتر باشد، آن را به لیست دیگری می‌بریم و درضمن مقدار رکورد تغییری نمی‌کند.

۱۵ گزینه (۴)

Pre: a b c d e f g

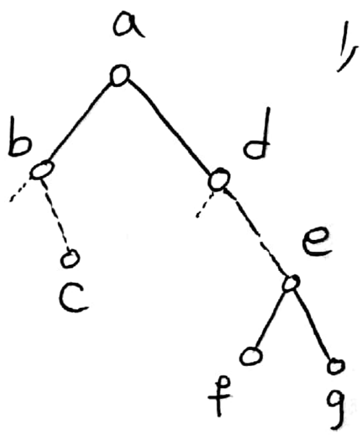
post: c b f g e d a

a ریشه است و اگر BOX های چپ و راست را مقایسه کنیم معلوم می‌شود c,b در BOX چپ و سایر گره‌ها در BOX راست قرار دارند:

Pre: a $\boxed{bc}$ $\boxed{defg}$

Post: $\boxed{cb}$ $\boxed{fged}$ a

به همین ترتیب در زیردرخت های چپ و راست، کار را ادامه می‌دهیم. هرگاه گره‌ای دیدیم که یکی از فرزندان آن NIL است بهتر است فرزند چپ را NIL در نظر بگیریم تا اندیس بزرگتر 2i+1 در آرایه ایجاد شود.



مثلاً d فقط یک زیردرخت دارد. این زیردرخت را در سمت راست آن قرار دادیم تا اندیس 2i+1 را ایجاد کنیم. d در i=۳ قرار دارد. e را عمداً در 2i+1=۷ فرض کردیم.

۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵
a	b	d		c		e							f	g

c می‌تواند در ۴ یا ۵ باشد
ما ۵ را انتخاب کردیم.
(فرزند راست)

فرزندان e

جواب = ۱۵ − ۷ = ۸

۱۶

گزینه (۱) [یک تمرین از CLRS]

تفاوتی ندارد که فکر کنید منظور طراح سؤال n درج و n حذف بوده یا آن که تعداد اعمال در مجموع n باشد.

فرض کنید مثلاً $\frac{n}{2}$ درج و پس $\frac{n}{2}$ حذف (استخراج) انجام دهیم و تعداد کل دستورات، n باشد:

$O(1)$: push(x_1)

$O(1)$: push(x_2)

⋮

$O(1)$: push($x_{\frac{n}{2}}$)

[ابتدا همه از S_1 به S_2 می‌آیند پس x_1 خارج می‌شود] : pop(S_2)

$O(\frac{n}{2}) + O(\frac{n}{2}) + O(1)$

$O(1)$: pop(S_2)

⋮

$O(1)$: pop(S_2)

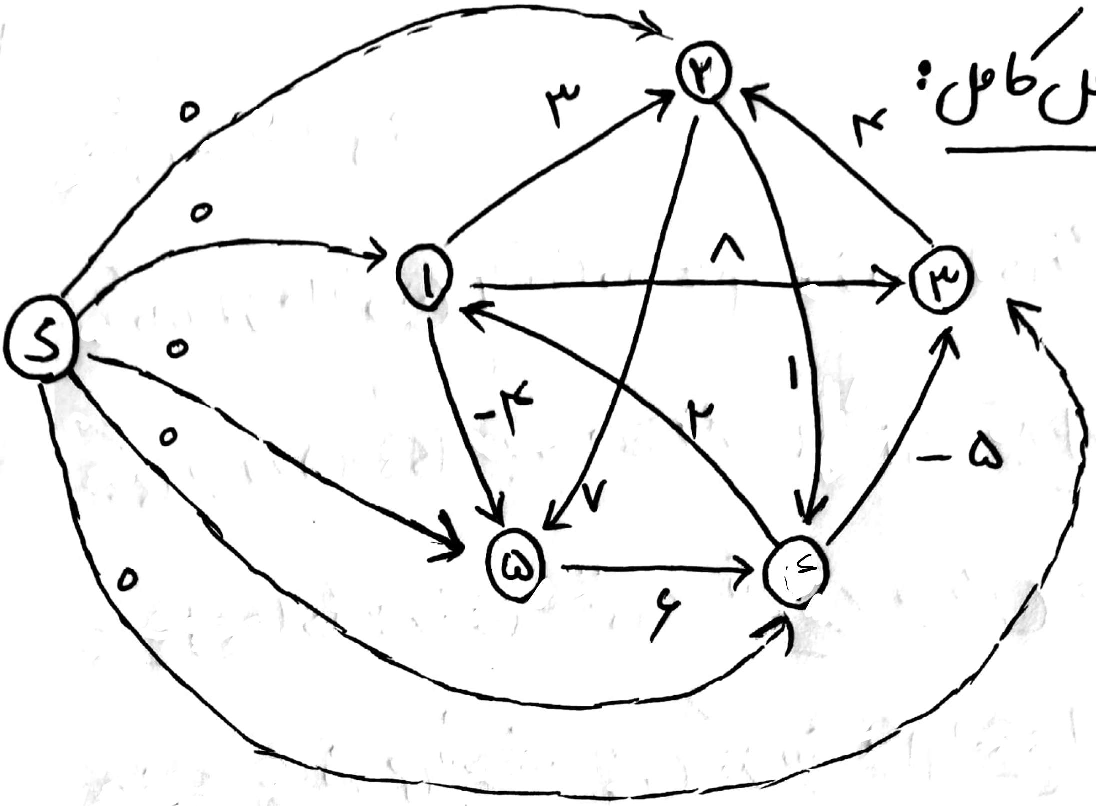
(مجموع زمان‌ها: $n + \frac{n}{2} + \frac{n}{2}$)

نتیجه‌ی اول: این ساختار یک صف است زیرا اولین ورودی اولین خروجی بود

نتیجه‌ی دوم: سرشکن $= \frac{n + \frac{n}{2} + \frac{n}{2}}{n} = 2 = O(1)$

۱۷ گزینه (۴)

روش حل کامل:



اضافه کردن گره s و یال‌های (s, u) با وزن صفر.

کوتاهترین مسیر (کم هزینه‌ترین) از گره ⓢ به ① همان مسیر مستقیم با هزینه‌ی صفر است. $F(1) = 0$

کوتاهترین مسیر از ⓢ به ⑤ مسیر $s \to 1 \to 5$ است که وزن آن $F(5) = -4$ است.

حالا در گراف جدید وزن‌ها به این صورت محاسبه می‌شود:

$$w'(1,5) = w(1,5) + F(1) - F(5)$$

$$= -4 + 0 - (-4) = 0$$

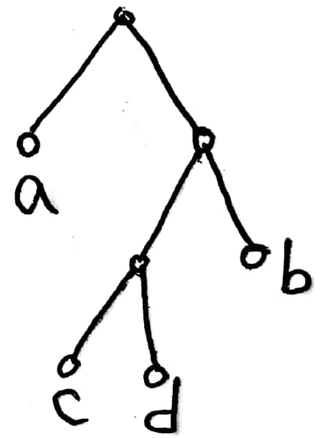
۱۸ گزینه (۳)

	c	d	b	a
f_i	۱	۳	۵	۶

	(cd)	b	a
f_i	۴	۵	۶

	a	((cd) b)
f_i	۶	۹

درخت هافمن = (a ((cd) b))



طول رشته حاصل $= \sum_{\text{برگ‌ها}}$ فراوانی × عمق

$= ۳(۱+۳) + ۲(۵) + (۶) = ۲۸$

۱۹ گزینه (۳)

می‌دانیم که تعداد مقایسه‌ها در این روش‌های مرتب‌سازی به این ترتیب است:

مرتب‌سازی درجی: $O(n^2)$ مرتب‌سازی شمارشی $O(n+m)$

مرتب‌سازی سریع $O(n\log n)$ و مرتب‌سازی ادغامی $O(n\log n)$.

حالا طبق صورت سؤال، انجام هر مقایسه، عملیاتی از مرتبه‌ی $O(K)$ است که K خود از مرتبه‌ی $O(\log n)$ است. بنابراین هر مقایسه به مرتبه‌ی زمانی $O(\log n)$ نیاز دارد پس همه‌ی موارد چهارگانه فوق باید در $\log n$ ضرب شوند:

مرتب‌سازی درجی تحت شرایط این مسئله: $O(n^2\log n)$

" شمارشی " " " " " : $O((n+m)\log n)$

مرتب‌سازی سریع " " " " " : $O(n\log^2 n)$

مرتب‌سازی ادغامی " " " " " : $O(n\log^2 n)$

در واقع ما همیشه فرض می‌کردیم یک مقایسه از مرتبه $O(1)$ باشه اما در این سؤال یک مقایسه خودش از مرتبه $O(\log n)$ است.

[فهم حالت کلی Merge sort را در کانال های بابان تمام کنید:]

۲۵ گزینه (۲)

در حالت کلی merge sort وقتی تقسیم را تا رسیدن به آرایه های با طول حداکثر m ادامه می دهیم و آنها را با روشی از مرتبه ی زمانی $f(m)$ مرتب می کنیم، مرتبه ی زمانی کل برابر است با:

$$T(n) = O\left(n \log \frac{n}{m} + \frac{n}{m} f(m)\right)$$

در این تست $m = \sqrt{n}$ است و $f(m) = m^2$ است. زیرا مرتبه ی زمانی مرتب سازی درجی برای m عدد، $O(m^2)$ است.

بنابراین:

$$T(n) = O\left(n \log \sqrt{n} + \sqrt{n}\,(\sqrt{n})^2\right)$$

$$= O(n\sqrt{n})$$

پیروز و سربلند باشید @abolfazlgilak