

۱ - زمان انجام هر یک از اعمال زیر در درخت دودویی جستجو شامل  $n$  عنصر چقدر است؟ (هر مورد ۰.۵ نمره)

- الف) درج در بهترین حالت  $O(1)$   
 ب) درج در بدترین حالت  $O(n)$  و یا  $O(\text{height})$   
 پ) پیدا کردن کوچک‌ترین عنصر درخت  $O(n)$  و یا  $O(\text{height})$   
 ت) پیدا کردن بزرگ‌ترین عنصر درخت  $O(n)$  و یا  $O(\text{height})$   
 ث) چاپ تمامی عناصر موجود در درخت از کوچک به بزرگ  $O(n)$   
 ج) حذف در بهترین حالت  $O(1)$   
 چ) حذف در بدترین حالت  $O(n)$  و یا  $O(\text{height})$   
 ح) حداقل ارتفاع درخت در بهترین حالت  $O(\log n)$   
 خ) حداکثر ارتفاع درخت در بدترین حالت  $O(n)$

۲ - زمان اجرای هر یک از توابع زیر را بر اساس ورودی  $N$  که به آن تابع داده شده است، بنویسید. (هر مورد ۱ نمره)

زمان اجرا	تابع
$O(N)$	<pre>public static int f1(int N) {     int x = 0;     for (int i = 0; i &lt; N; i++)         x++;     return x; }</pre>
$O(N^2)$	<pre>public static int f2(int N) {     int x = 0;     for (int i = 0; i &lt; N; i++)         for (int j = 0; j &lt; i; j++)             x++;     return x; }</pre>
$O(N!)$	<pre>public static int f3(int N) {     if (N == 0) return 1;     int x = 0;     for (int i = 0; i &lt; N; i++)         x += f3(N-1);     return x; }</pre>
$O(2^N)$	<pre>public static int f6(int N) {     if (N == 0) return 1;     return f6(N-1) + f6(N-1); }</pre>
$O(\log N)$	<pre>public static int f7(int N) {     if (N == 1) return 0;     return 1 + f7(N/2); }</pre>

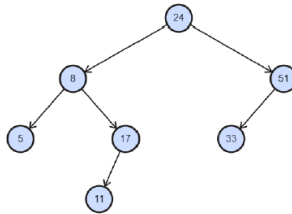
۱ - رابطه بازگشتی زیر را حل کنید: (۱ نمره)

$$T(n) = 7T\left(\frac{n}{3}\right) + n^2/3$$

با استفاده از قضیه اصلی داریم:  $a=7$ ,  $b=3$ , و  $d=2$ . چون  $2 > \log_3 7$ , بنابراین جواب  $T(n) = \Theta(n^2)$  خواهد بود.

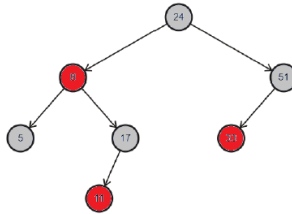
۲ - اعداد زیر را به ترتیب از چپ به راست در یک درخت دودویی جستجو درج می‌کنیم (ابتدا ۲۴، سپس ۵۱، ...):  
24, 51, 8, 5, 17, 33, 11

الف) پس از درج تمامی این عناصر، درخت حاصل را نمایش دهید. (۱ نمره)



ب) آیا می‌توان رنوس این درخت را به گونه‌ای با دو رنگ قرمز و سیاه رنگ کرده که درخت حاصل یک درخت قرمز-سیاه باشد؟ اگر جواب مثبت است، یک رنگ‌آمیزی ممکن را نمایش دهید. اگر جواب منفی است، چرا نمی‌توان این کار را کرد؟ (۱ نمره)

بله، می‌توان:



۱ - چهار عدد ۱، ۲، ۳، و ۴ به ترتیب به یک پشته باید با استفاده از عمل «درج» وارد شوند. از طرفی، در هر زمانی می‌توان با استفاده از عمل «حذف»، عنصری که فعلاً در روی پشته است را خارج کرد. مشخص کنید که برای اینکه ترتیب اعداد خارج شده از پشته به صورت هر یک از موارد زیر باشد، با چه ترتیبی باید عملیات درج و حذف انجام شود.

الف) ابتدا ۴، سپس ۳، پس از آن ۲، و در پایان ۱

درج، درج، درج، حذف، حذف، حذف، حذف

ب) ابتدا ۱، سپس ۲، پس از آن ۳، و در پایان ۴

درج، حذف، درج، حذف، درج، حذف، درج، حذف

ج) ابتدا ۳، سپس ۴، پس از آن ۱، و در پایان ۲

غیر ممکن است. برای اینکه ۳ ابتدا خارج شود، باید پس ابتدا ۳ بار درج انجام دهیم و پس از آن اولین حذف را انجام دهیم. از اولین درج، حذف انجام دهیم. پس از آن نیز باید یک درج و سپس حذف انجام دهیم تا ۴ خارج شود. حال در پشته اعداد ۱ و روی آن ۲ داریم و بنابراین با حذف بعدی، ۲ خارج خواهد شد و نه ۱.

۳ - در ساختمان داده صف که با لیست پیوندی یکطرفه پیاده سازی شده است، علاوه بر اضافه کردن یک عنصر به انتهای لیست و

حذف عنصر از ابتدای لیست، می‌خواهیم عملیات حذف عنصر از انتهای لیست را نیز پیاده سازی کنیم. زمان اجرای این عمل

جدید چقدر خواهد بود؟ چرا؟

برابر  $\Theta(n)$  خواهد بود زیرا برای حذف آخرین عنصر، نیاز به یافتن عنصر ماقبل آن داریم (تا هم عنصر بعدی آن را به مقدار هیچ تغییر دهیم و هم اشاره‌گر آخرین عنصر را به آن ارجاع دهیم). برای یافتن عنصر ماقبل آخر، نیاز است از ابتدا شروع کرده و تا آخر برویم که نیاز به زمان  $\Theta(n)$  دارد.

۴ - روز اول سال تحصیلی برای سجاد است. در کلاس سجاد،  $n$  دانش‌آموز وجود دارند. قرار است که دانش‌آموزان به ترتیب قد

مرتب شوند. سجاد خیلی عجله دارد می‌خواهد سریع‌تر بداند ۵ نفری که در صف مرتب شده جلوی او خواهند بود، چه کسانی

هستند. برای او توضیح دهید که چگونه در زمان  $O(n)$  می‌تواند این افراد را پیدا کند؟ (۲ نمره)

او باید ابتدا تمامی افرادی را که از او قد کوتاه‌تری دارند را پیدا کند. این کار با یک‌بار پردازش لیست و افراز آن بر مبنای قد

خودش (مشابه الگوریتم افراز مرتب‌سازی سریع) در زمان  $O(n)$  انجام می‌شود. سپس بین این افراد، ۵ بار کوتاه‌ترین فرد را پیدا

کرده و کنار می‌گذارد. هربار اجرای این کار  $O(n)$  طول کشیده و در کل  $O(n)$  زمان لازم خواهد داشت.

۶ - فرض کنید که خداوند هر روز صبح لیستی مرتب‌شده از کد ملی افرادی را که فردا زنده هستند را به عزرائیل می‌دهد. عزرائیل باید با مقایسه این لیست با لیست افراد زنده امروز که صبح روز قبل دریافت کرده است، لیست افرادی را که باید امروز به ملاقتشان برود را مشخص کند. اگر لیست دریافتی امروز شامل  $n$  کد ملی و لیست دریافتی روز گذشته شامل  $m$  کد ملی باشد، چطور می‌تواند این کار را در زمان  $O(m+n)$  انجام دهد؟ (۱ نمره)

او ابتدا کد ملی ابتدای هر دو لیست را با هم مقایسه می‌کند. اگر کد ملی یکسان بود، یعنی آن فرد روز قبل زنده بوده است و امروز هم زنده بوده است، پس آن را از هر دو لیست حذف کرده و ادامه می‌دهد. اگر کد ملی نفر لیست روز قبل کوچک‌تر بود، یعنی آن فرد روز قبل زنده بوده است و امروز نیست، پس آن فرد را به لیست خود اضافه کرده و از لیست روز قبل حذف می‌کند. در غیر این صورت، کد ملی نفر اول لیست امروز کوچک‌تر خواهد بود. این به این معنا است که این فرد امروز متولد شده است. پس آن را از لیست امروز حذف کرده و ادامه می‌دهد. در این الگوریتم، در هر مرحله یک نفر از یکی از دو لیست (و یا از هر دو لیست) حذف می‌شود. در نتیجه، پس از  $m+n$  گام، هر دو لیست خالی شده و کار تمام می‌شود. پس زمان اجرای آن  $O(m+n)$  است.

۴ - می‌خواهیم محل دانشجویان در سالن امتحانات برای آزمون پایان‌ترم را بر اساس شماره دانشجویی مشخص کنیم. برای این کار، یک درخت دودویی جستجو شامل دانشجویان بر اساس شماره دانشجویی ایجاد شده است. البته با توجه به چپ دست بودن برخی دانشجویان، قرار است که دانشجویان چپ دست در یک ردیف جداگانه جای داده شوند. چگونه می‌توان از روی این درخت، دو لیست مرتب شده بر اساس شماره دانشجویی از دانشجویان چپ دست و راست دست در زمان  $O(n)$  ایجاد کرد؟ (۲ نمره)

ابتدا در زمان  $O(n)$  و با استفاده از پیمایش میان‌ترتیب، یک لیست مرتب‌شده از کلیه دانشجویان ایجاد می‌کنیم. سپس این لیست را یکبار پیمایش کرده و به دو لیست دانشجویان چپ‌دست و راست‌دست افزایش می‌کنیم. این کار نیز در زمان  $O(n)$  ممکن است زیرا به ازای هر دانشجو، بر حسب اینکه چپ دست است و یا راست دست، او را در یکی از لیست قرار می‌دهیم.

۸ - سجاد مسئول تقسیم کردن باقی‌مانده آجیل‌های شب یلدا بین افراد فامیل شده است. او بدون استفاده از ترازو، آجیل‌ها را به  $n$  بسته تقسیم کرد. بعداً که افراد برای تحویل گرفتن آجیل‌ها آمدند، میزان انتظار افراد برای دریافت آجیل با هم تفاوت داشت. سجاد می‌خواهد ببیند که آیا می‌تواند به گونه‌ای بسته‌ها را بین افراد تقسیم کند که هر کسی میزان آجیل دریافتی‌اش از میزان انتظارش بیشتر باشد یا نه؟ اگر تعداد افراد  $n$  نفر، میزان آجیل موجود در بسته  $i$ ام برابر با  $w_i$  و میزان آجیل انتظاری فرد  $i$ ام هم  $e_i$  باشد، چگونه می‌توان در زمان  $O(n \lg n)$  بررسی که آیا این کار ممکن است یا نه، و اگر بله، یک جواب ممکن برای آن را پیدا کرد. (۴ نمره)

ابتدا وزن بسته‌ها و همچنین میزان انتظار افراد را مرتب می‌کنیم. برای اینکار به  $O(n \lg n)$  نیاز داریم. اگر لیست مرتب‌شده وزن‌ها به ترتیب  $w_1 \leq w_2 \leq \dots \leq w_n$  و لیست مرتب‌شده انتظارات  $e_1 \leq e_2 \leq \dots \leq e_n$  باشد، آنگاه اینکار ممکن است اگر برای تمام آنها داشته باشیم  $w_i \geq e_i$ . در این حالت، کافی است بسته  $i$ ام را به فرد  $i$ ام بدهیم. واضح این که این یک جواب درست است. اما اگر برای یک  $i$  داشته باشیم  $w_i < e_i$  آنگاه مسئله جواب ندارد، زیرا هیچ‌یک از بسته‌های  $w_1$  تا  $w_i$  را نمی‌توان به افراد  $i$  تا  $n$  داد زیرا انتظار تمام آن‌ها حداقل  $e_i$  است که آن هم از تمام این بسته‌ها بزرگ‌تر است. در نتیجه اگر مسئله جواب داشته باشد، باید بسته‌های  $i+1$  تا  $n$  را بین اعضای  $i$  تا  $n$  تقسیم کند که غیر ممکن خواهد بود.