



بسمه تعالی  
ساختمان داده ها و الگوریتم ها  
امتحان پایان ترم  
چهارشنبه ۲۴ خرداد ۱۴۰۲



مدت آزمون ۹۰ دقیقه

شماره دانشجویی:

نام و نام خانوادگی:

۰ - سوال امتیازی: جذابترین لطیفه‌ای که در طول ترم سر کلاس شنیدید را بنویسید. (۱ نمره)

۱ - در جدول زیر، ستون سمت راست آرایه ورودی به الگوریتم مرتب‌سازی و ستون سمت چپ، آرایه مرتب شده در انتهای مرتب‌سازی را نمایش می‌دهد. هر یک از ستون‌های میانی، محتوای آرایه در طی فرآیند مرتب‌سازی توسط یکی از الگوریتم‌های مرتب‌سازی سریع، ادغامی، درجی و انتخابی را نشان می‌دهند. مشخص کنید که هر ستون متعلق به فرآیند مربوط به کدام روش مرتب‌سازی است. (هر مورد ۱ نمره)

ورودی	درجی	ادغامی	انتخابی	سریع	مرتب شده
navy	blue	blue	bark	coal	bark
plum	coal	coal	blue	jade	blue
coal	corn	gray	cafe	blue	cafe
jade	gray	jade	coal	gray	coal
blue	jade	lime	corn	corn	corn
pink	lime	mint	dusk	bark	dusk
rose	mint	navy	navy	dusk	gray
gray	navy	pink	plum	herb	herb
teal	pink	plum	jade	cafe	jade
ruby	plum	rose	pink	leaf	leaf
mint	rose	ruby	rose	navy	lime
lime	ruby	teal	gray	plum	mint
silk	silk	bark	teal	pink	mist
corn	teal	corn	ruby	rose	navy
bark	bark	dusk	mint	teal	palm
wind	wind	leaf	lime	ruby	pine
dusk	dusk	silk	silk	mint	pink
leaf	leaf	wind	wind	lime	plum
herb	herb	cafe	leaf	silk	rose
sage	sage	herb	herb	wind	ruby
cafe	cafe	mist	sage	sage	sage
mist	mist	palm	mist	mist	silk
pine	pine	pine	pine	pine	teal
palm	palm	sage	palm	palm	wind



مدت آزمون ۹۰ دقیقه

شماره دانشجویی:

نام و نام خانوادگی:

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>
$O(N \log N)$	<pre>public static int f4(int N) {     if (N == 0) return 0;     return f4(N/2) + f1(N) + f4(N/2); }</pre>
$O(N)$	<pre>public static int f5(int N) {     int x = 0;     for (int i = N; i &gt; 0; i = i/2)         x += f1(i);     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>



نام و نام خانوادگی:

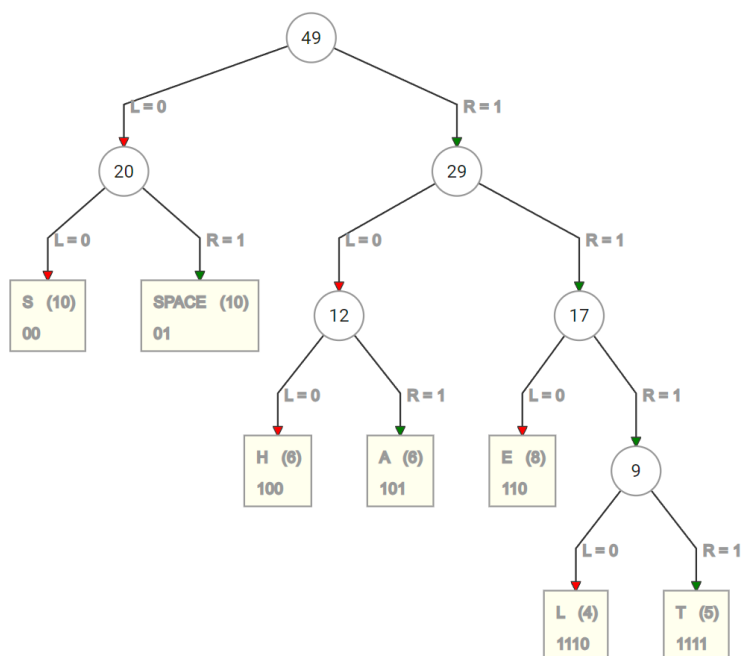
شماره دانشجویی:

مدت آزمون ۹۰ دقیقه

۳ - می‌خواهیم برای فشرده‌سازی عبارت زیر از روش کد هافمن استفاده کنیم (فاصله را نیز یک حرف در نظر بگیرید):

SHE SELLS SEA SHELLS THAT SHE HAS AT THE EAST SEA

برای این کار، ابتدا جدول فراوانی هر حرف (که شامل فاصله به عنوان یک حرف نیز می‌شود) را کامل کنید (۲ نمره). سپس درخت هافمن را رسم کنید (۲ نمره). در پایان نیز کد متناظر با هر حرف را در جدول مشخص کنید (۲ نمره).



کد هافمن	فراوانی	حرف
101	6	A
110	8	E
100	6	H
1110	4	L
00	10	S
1111	5	T
01	10	SPACE

۴ - یکی از محلات توریستی شهر قزوین، خیابان سفیر امید معروف به رودخانه‌ی بازار در محله نواب است. نکته دیدنی این محله، خانه‌های رنگی آن است. شهردار منطقه ۶ که در تعطیلات اخیر به قزوین رفته بود، پس از مشاهده آن، تصمیم می‌گیرد که جهت افزایش شادی دانشجویان دانشگاه امیرکبیر،  $n$  ساختمان شمال خیابان رشت را نیز با استفاده از سه رنگ قرمز، سبز، و آبی رنگ کند. پس از یک بررسی اولیه و با توجه به قیمت هر یک از این رنگ‌ها و نمای خانه‌ها، هزینه لازم برای رنگ کردن هر خانه به هر کدام از این رنگ‌ها آماده شده است (هزینه رنگ کردن خانه آبی به رنگ قرمز، آبی، و سبز به ترتیب برابر بی  $CR[i]$ ،  $CB[i]$  و  $CG[i]$  است). روشی ارائه دهید که کم‌ترین هزینه لازم برای رنگ کردن ساختمان‌ها را به گونه‌ای که هیچ دو ساختمان مجاور هم‌رنگ نباشند. زمان الگوریتم خود را هم بیان کنید. (۴ نمره)

سه مقدار  $R[i]$ ،  $G[i]$  و  $B[i]$  را به این صورت تعریف می‌کنیم که هر یک کمترین هزینه رنگ کردن  $i$  خانه اول باشد به گونه‌ای که خانه آبی قرمز، سبز، و یا آبی باشد. آنگاه رابطه محاسبه بازگشتی این سه تابع به صورت زیر خواهد بود:

$$R[i+1] = CR[i] + \min \{G[i], B[i]\} \quad B[i+1] = CB[i] + \min \{G[i], R[i]\} \quad G[i+1] = CG[i] + \min \{R[i], B[i]\}$$

همچنین خواهیم داشت که  $R[0]=B[0]=G[0]$ . به این ترتیب، هر یک از این مقادیر در زمان  $O(1)$  از روی مقادیر قبلی محاسبه می‌شوند. پس در کل، در زمان  $O(n)$  هر سه تابع برای تمام مقادیر  $i$  از ۱ تا  $n$  محاسبه می‌شود. جواب مسئله نیز  $\max\{R[n], G[n], B[n]\}$  است زیرا خانه آخر یکی از سه رنگ را خواهد داشت.



مدت آزمون ۹۰ دقیقه

شماره دانشجویی:

نام و نام خانوادگی:

۵ - رضا یک شلنگ آب دست دوم از روی دیوار پیدا کرده و خرید. وقتی که دیروز می خواست از آن استفاده کند، متوجه شد که  $n$  جای آن سوراخ است. او محل دقیق سوراخ ها نسبت به ابتدای شلنگ را اندازه گیری کرد که تشکیل یک لیست از  $n$  عدد حقیقی مثبت را داد. برای پنچرگیری شلنگ، او باید از وصله های ۵ سانتی متری استفاده کند. در نتیجه، ممکن است بتواند برای چندین سوراخ نزدیک به هم تنها از یک وصله استفاده کند. برای کمک به او روشی برای محاسبه کمترین تعداد وصله مورد نیاز طراحی کنید. روش خود را به صورت متنی و یا شبه کد بیان کرده، زمان اجرای آن را تحلیل کرده و درستی آن را اثبات کنید (۶ نمره). ابتدا محل سوراخ ها را مرتب می کنیم. سپس از اولین سوراخ شروع کرده و یک وصله برای آن استفاده می کنیم. این برچسب این سوراخ و تمامی سوراخ هایی را که کمتر از ۵ سانتی متر از آن فاصله دارند را خواهد پوشاند. بنابراین در آرایه مرتب شده به قدری جلو می رویم که یا تمام شود و یا به سوراخی برسیم که از این سوراخ بیش تر از ۵ سانتی متر فاصله دارد. در آن صورت، یک وصله دیگر برای آن سوراخ استفاده کرده و به همین نحو ادامه می دهیم.

زمان اجرا: مرتب سازی به زمان  $O(n \log n)$  نیاز دارد. در ادامه، برای هر سوراخ زمان  $O(1)$  مصرف می کنیم زیرا فقط چک می کنیم که از محل آخرین سوراخی که برای آن یک وصله جدید استفاده کردیم بیش از ۵ سانتی متر فاصله دارد یا خیر. و اگر بله، یک وصله جدید بر می داریم. بنابراین کل زمان این پیمایش نیز  $O(n)$  خواهد بود و زمان کلی نیز  $O(n \log n)$  خواهد بود. درستی: جواب بهینه را در نظر بگیرید. برای پوشاندن اولین سوراخ حتما یک وصله نیاز است. می توان این وصله را آنقدر جلو برد که این سوراخ در ابتدای آن باشد. به این ترتیب، این وصله تمامی سوراخ هایی را که کمتر از ۵ سانتی متر از سوراخ اول فاصله دارند را نیز خواهد پوشاند. و بقیه سوراخ ها نیز باید توسط سایر وصله ها پوشانده شده باشند. به عبارتی، اولین وصله این جواب تغییر یافته مشابه اولین وصله ای است که روش ما استفاده می کند و سایر وصله ها نیز بقیه سوراخ های باقی مانده را باید بپوشانند. پس اولین انتخاب ما اشتباه نیست و با استقرا ثابت می شود که الگوریتم ما در پایان به جواب بهینه خواهد رسید.

۶ - یک جدول  $n \times n$  از اعداد داریم. روشی ارائه کنید که در زمان  $O(n^2 \log n)$  بزرگترین عددی را که در تمام سطرها تکرار شده است را بیابد (و یا عدم وجود عددی که در تمام سطرها تکرار شده باشد را بیان کند). (۳ نمره)

برای هر سطر، یک درخت قرمز-سیاه درست کرده و تمامی اعداد آن سطر را در آن درج می کنیم. این کار برای هر سطر  $O(n)$   $\log n$  و در کل  $O(n^2 \log n)$  طول خواهد کشید. سپس سطر اول را مرتب می کنیم که آن هم  $O(n \log n)$  طول می کشد. در ادامه، از بزرگترین عدد سطر اول شروع کرده و بررسی می کنیم که آیا این عدد در تمام سطرها دیگر وجود دارد یا نه. چک کردن در هر سطر  $O(\log n)$  نیاز دارد و حداکثر در زمان  $O(n \log n)$  مشخص می شود که آیا این عدد در تمام سطرها دیگر هم است یا نه. اگر بود، جواب مسئله همین عدد است و کار تمام است. در غیر این صورت، به سراغ بزرگترین عدد بعدی می رویم. چون در کل  $n$  عدد در سطر اول داریم، این بخش هم  $O(n^2 \log n)$  طول خواهد کشید. اگر هیچ یک از اعداد سطر اول در تمام سطرها دیگر نبودند هم جواب مسئله نه است.