باسمه تعالی

شانزدهمین دوره‌ی آموزشی المپیاد کامپیوتر

امتحان میان‌ترم درس الگوریتم

شنبه ۱۴ مردادماه ۱۳۸۵

نصیری‌شرق، سیّدی — فرصت: ۱۵۳ دقیقه

مسئله‌ی صفر: چند نکته‌ی خیلی مهم! ..... ۳ دقیقه

توصیه می‌شود قبل از شروع امتحان به نکات زیر توجّه کنید:

- برای این قسمت/مسئله ۳ دقیقه وقت درنظر گرفته شده‌است. این مدّت را صرف خواندن همین نکات کنید.
- فرصت این آزمون (در مقایسه با تعداد و حجم سؤالات) ممکن است کم به‌نظر برسد. از این رو، حتماً برای زمان خود برنامه‌ریزی کنید؛ چون وقت امتحان به‌هیچ‌وجه اضافه نخواهد شد.
- سعی کنید پاسخ‌های خود را «مختصر» و «دقیق» بنویسید. بدیهی‌ست که بیان نادقیق یا بدون اثبات یک پاسخ/الگوریتم ممکن است هیچ نمره‌ای نداشته باشد. «اختصار»، عامل کلیدی در مدیریت زمان و «دقّت» عامل کلیدی در گرفتن نمره می‌باشد!
- در سؤالاتی که از شما یک الگوریتم به زبان CLRS خواسته شده، از نوشتن الگوریتم به زبان‌های دیگر خودداری کنید. در سؤالات سوم، هفتم و هشتم نیازی به کد الگوریتم نیست؛ توصیف دقیق رویه به فارسی کفایت می‌کند.

مسئله‌ی اوّل: هفته‌بازار ٪ غ‌ص‌ص‌غ‌ص‌غ‌غ ..... ۳۵ نمره

در هر یک از موارد زیر، درستی یا نادرستی عبارت ذکر شده را مشخّص کنید و دلیل خود را به‌دقّت توضیح دهید. توجّه کنید که پاسخ خالی و بدون ذکر دلیل، نمره‌ای در پی نخواهد داشت.

الف) پیدا کردن عنصر بیشینه در یک Min-Heap از زمان $\Omega(\lg n)$ است.

ب) بنابر قضیه‌ی اصلی، جواب رابطه‌ی بازگشتی $T(n) = ۴T(\frac{n}{۵})+\lg(n!)$ برابر است با $T(n) = \Theta(n\lg n)$.

ج) یک Heap با n عنصر را می‌توان در زمان $O(n)$ به یک درخت دودویی جستجو تبدیل کرد.

د) عدد ثابتی مانند $n_۰ \geq ۱$ وجود دارد که برای هر $n \geq n_۰$، آرایه‌ای به طول n وجود دارد که روی آن آرایه، الگوریتم Insertion Sort سریع‌تر از Merge Sort عمل می‌کند.

ه) برای گراف وزن‌دار $G = (V, E)$، درخت M یک زیردرخت فراگیر کمینه است، اگر و فقط اگر برای هر دو رأس دلخواه $v_۱$ و $v_۲$، وزن مسیر بین آن دو در M، کمینه‌ی وزن مسیرهای بین آن دو در G باشد.

و) یک درخت قرمز—سیاه با ۱۰۲۴ گره، حداقل یک گره‌ی قرمز دارد.

ی) عمل درج در درخت قرمز—سیاه دارای خاصیت جابه‌جایی است. بدین معنی که درج عنصر x و سپس عنصر y همان درختی را ایجاد می‌کند که درج y و سپس x می‌سازد.

مسئله‌ی دوم: ۱۱۸ مغناطیسی ٪ ددج ..... ۱۰ نمره

در دفترچه‌ی تلفن معمولی (که در انتهای تقویم‌ها و سررسیدها وجود دارد)، از چه داده‌ساختاری برای نگه‌داری شماره‌ی تلفن‌ها استفاده می‌شود؟
زمان متوسّط جستجوی ناموفق یک نام در این داده‌ساختار چه‌قدر است؟ اثبات کنید.

مسئله‌ی سوم: الگوریتم‌کمکی ٪ ب‌ب‌خ‌خ‌ب ..... ۱۵ نمره

از بین الگوریتم‌های مرتّب‌سازی Heap-Sort، Merge-Sort، Quick-Sort، Insertion-Sort و Bucket-Sort کدام‌ها (مستقل از پیچیدگی زمان اجرا) می‌توانند (با تغییرات جزئی در صورت نیاز) به عنوان الگوریتم کمکی برای الگوریتم Radix-Sort به‌کار گرفته شوند؟

در صورت تأیید هر الگوریتم، اصلاحات ضروری آن را ذکر کرده و در صورت رد، دلیل خود را (در صورت نیاز با ذکر مثال) توضیح دهید.

مسئله‌ی چهارم: مرتّب‌سازی درغامی ..... ۱۵ نمره

الگوریتم Inserge-Sort(A) که هدف‌ش مرتّب‌سازی آرایه‌ی A است، بدین صورت است:

```
Inserge-Sort(A)
    ▷ Sorts array A by Insertion & Merge
    n ← Length(A);
    k ← Random(1, ⌊n/2⌋);
    Insertion-Sort(A[1..k]);
    Inserge-Sort(A[k+1..n]);
    Merge(A[1..k], A[k+1..n]);
```

که منظور از $A[x..y]$، زیرآرایه‌ی شامل $A[x]$ تا $A[y]$ است و تابع Random(x, y) یک عدد تصادفی در بازه‌ی $[x,\ y]$ برمی‌گرداند.

رابطه‌ی بازگشتی زمان اجرای این الگوریتم را بر حسب n نوشته و زمان اجرای متوسّط آن را صریحاً تحلیل کنید.

مسئله‌ی پنجم: پسرخاله ..... ۱۵ نمره

در یک درخت دودویی جست‌جو، پسرخاله‌ی گره‌ی x، گره‌ای است که در صورتی که عناصر درخت را مرتّب کنیم، دقیقاً قبل از x قرار بگیرد. برای مثال در ددج‌ای که اعداد ۱ تا n در آن (به هر ترتیبی) قرار گرفته باشند، پسرخاله‌ی گره‌ای با مقدار $1 < i \leq n$، گره‌ی با مقدار $i + 1$ است.

الگوریتمی با اعلان Tree-Cousin(x) به‌زبان CLRS بنویسید که پسرخاله‌ی یک گره را برگرداند. دقّت کنید که x یک گره (و نه مقدار آن) است و خروجی الگوریتم هم باید یک گره (و نه یک مقدار) باشد. در صورتی که آن گره پسرخاله نداشت (عنصر کمینه بود)، NIL باید برگردانده شود.

توجّه کنید که در این مسئله، درصورتی که طول شبه‌کد شما (بدون شماردن خط اعلان و کامنت‌ها) از ۱۰ خط استاندارد CLRS تجاوز کند، هیچ نمره‌ای نخواهید گرفت.

مسئله‌ی ششم: **کپه‌ی بازگشتی** ..... ۳۰ نمره

رویه‌ی Build-Heap را برای تبدیل آرایه‌ی $A[۱..n]$ به یک هیپ در نظر بگیرید:

```
BUILD-HEAP(A)
    for i ← ⌊Length(A)/2⌋ downto 1
        do Bubble-Down(A, i);
```

الف) رویه‌ی فوق را به صورت رویه‌ی بازگشتی Recursive-Build-Heap(A, i) به زبان CLRS بازنویسی کنید، طوری که A[i] ریشه‌ی «زیر—هیپ»ی است که قرار است ساخته شود. برای تبدیل کل آرایه به یک هیپ باید رویه را به‌صورت Recursive-Build-Heap$(A, ۱)$ فرا بخوانیم.

ب) زمان اجرای الگوریتم خود را در بدترین حالت، به‌صورت یک رابطه‌ی بازگشتی بنویسید.

ج) رابطه‌ی بازگشتی به‌دست آمده را با استفاده از قضیه‌ی اصلی حل کنید.

مسئله‌ی هفتم: **محاسبه‌ی جملات** ..... ۳۰ نمره

دنباله‌ی f روی اعداد طبیعی بدین صورت تعریف می‌شود:

$$\begin{cases} f_i = d_i & ۱ \leq i \leq k \\ f_i = \sum_{j=۱}^{k} c_j \times f_{n-j} & i > k \end{cases}$$

که در آن $k \geq ۱$، d_i ها و c_i ها اعداد ثابت صحیحی هستند.

الگوریتمی از زمان $O(\lg n)$ بدهید که جمله‌ی n اُم این دنباله (f_n) را محاسبه کند.

مسئله‌ی هشتم: **چند تا؟** ..... ۲۰ نمره

گراف ساده و بدون وزن $G = (V, E)$ را در نظر بگیرید. دو رأس u و v از این گراف داده شده است. می‌خواهیم تعداد کوتاهترین مسیرها بین این دو رأس را پیدا کنیم. الگوریتمی با زمان اجرای $O(|V| + |E|)$ برای این کار ارائه کنید. الگوریتم خود را تحلیل و اثبات کنید.

مسئله‌ی نهم: **اگه!** ..... ۳۰ نمره

گراف ساده و وزن دار $G = (V, E)$ را در نظر بگیرید. وزن همه‌ی یال‌های این گراف نامنفی است. می‌دانیم که M زیردرختی فراگیر با کمترین وزن در این گراف است. همچنین می‌دانیم که P کوتاه‌ترین مسیر بین دو رأس u و v است.

حالا فرض کنید که به جای وزن هر یال، مجذور وزن آن را قرار می‌دهیم؛ مثلاً، اگر وزن یالی ۳ بوده وزن آن را ۹ می‌کنیم.

الف) آیا در گراف جدید، همان P قبلی لزوماً کوتاه‌ترین مسیر بین u و v است؟

ب) آیا در گراف جدید، همان M قبلی لزوماً زیردرخت فراگیر با کمترین وزن است؟

در هر مورد، اگر جواب مثبت است آن را اثبات کنید و اگر جواب منفی است یک مثال نقض ارائه کنید.

«موفّق باشید»