

به نام خدا

سوالات پایان ترم درس ساختمان داده ها (۱۲ نمره، ۱۲۰ دقیقه)

نکته ۱: در مورد سوالاتی که از شما الگوریتم خواسته شده، نیازی به نوشتن کد نیست و روش کار را شرح دهید کافی خواهد بود.

نکته ۲: زمان تخمینی لازم برای هر سوال مشخص شده است.

سوال یک: روابط بازگشتی (۳ نمره) ۲۰ دقیقه

الف: کدام یک از شرایط برای مقادیر صحیح و مثبت  $a_1$  و  $a_k$  تضمین میکند که  $\theta(n)$  پاسخ رابطه بازگشتی زیر باشد؟ (۰٫۷۵)

$$T(n) = \sum_{i=1}^k T\left(\frac{n}{a_i}\right) + \theta(n)$$

د:  $\sum_{i=1}^k \frac{1}{a_i} = 1$

ج:  $\sum_{i=1}^k \frac{1}{a_i} > 1$

ب:  $\sum_{i=1}^k \frac{1}{a_i} < 1$

الف:  $\sum_{i=1}^k \frac{1}{a_i} = k$

ب: روابط زیر را حل کنید (۱٫۵).

$$T(n) = 3T(\sqrt{n}) + \lg(n)$$

$$T(n) = T\left(\frac{n}{2}\right) + T(\sqrt{n}) + n \quad T(1) = 1, \quad T(2) = 2$$

ج: فرض کنید  $a$  و  $b$  نمایش دو عدد صحیح مثبت باشند. فرض کنید تابع  $Q$  به شکل زیر به صورت بازگشتی تعریف شده است. مقدار  $Q(5861, 7)$  برابر است با (۰٫۷۵):

$$Q(a, b) = \begin{cases} 0 & \text{if } a < b \\ Q(a - b, b) + 1 & \text{if } b \leq a \end{cases}$$

سوال دو: مرتب‌سازی (۱ نمره) ۱۰ دقیقه

در گونه‌ای جدید از مرتب‌سازی سریع آرایه‌ای  $n$  عضوی،  $2\sqrt{n} + 1$  عنصر اول را انتخاب کرده با یک الگوریتم ساده مثل مرتب‌سازی درجی، این عناصر را مرتب و سپس عنصر میانه‌ی آن‌ها را بدست می‌آوریم. این عنصر را محور قرار داده و بقیه الگوریتم مرتب‌سازی را بر روی  $A$  اجرا میکنیم. کدام یک از رابطه‌های زیر، زمان اجرای این الگوریتم در بدترین حالت را نشان میدهد؟ **علت انتخاب خود را بیان کنید.**

الف:  $T(n) \leq T(n - \sqrt{n}) + O(\sqrt{n})$

ب:  $T(n) \leq T(2\sqrt{n}) + T(n - 2\sqrt{n}) + O(\sqrt{n})$

ج:  $T(n) \leq T(\sqrt{n}) + T(n - \sqrt{n}) + O(\sqrt{n})$

د:  $T(n) \leq 2T(n - \sqrt{n}) + O(\sqrt{n})$

سوال سه: جمع موازی (۵/۰ نمره) ۵ دقیقه

یک کامپیوتر با  $n/2$  پردازنده مفروض است. اگر بخواهیم با استفاده از این کامپیوتر اعداد موجود در یک آرایه  $n$  عنصری را جمع نماییم، زمان مورد نیاز چیست؟

الف:  $O(\log_{10} n)$

ب:  $O(\log_2 n)$

ج:  $O(n \log_{10} n)$

د:  $O(n/2)$

سوال چهار: درخت (۵/۱ نمره) ۲۰ دقیقه

الف: می‌دانیم که در یک درخت دودویی، سطح یا عمق یک گره برابر با طول مسیر از آن گره تا ریشه است. ارتفاع درخت هم بزرگترین سطح گره‌ها در آن درخت است. پهنای یک درخت دودویی را برابر با بیشترین تعداد گره‌های هم سطح در آن معرفی می‌کنیم. آیا درخت دودویی با  $n$  گره و ارتفاع و پهنای زیر وجود دارد؟ (۱)

۱: ارتفاع  $\theta(n)$  و پهنای ۱

۲: ارتفاع  $\theta(\log n)$  و پهنای  $\theta(n)$

۳: ارتفاع  $\theta(n)$  و پهنای  $\theta(n)$

۴: ارتفاع  $\theta(\log n)$  و پهنای  $\theta(\sqrt{n})$

برای هر کدام از گزینه‌ها، بله یا خیر را مشخص کنید و دلیل یا مثال بیاورید.

ب: بزرگ‌ترین عنصر یک مین-هیپ با عناصر متفاوت را به صورت کارا با چند مقایسه بین عناصر آن می‌توان به دست آورد؟ (۵/۰)

الف:  $\lfloor \log n \rfloor - 1$

ب:  $n - 1$

ج:  $\lfloor n/2 \rfloor - 1$

د:  $\lfloor (n + 1)/2 \rfloor$

سوال پنج: ادغام (۵/۱ نمره) یکی از دو مورد را پاسخ دهید. ۲۰ دقیقه

الف: دو مین-هیپ با نام‌های  $H_1$  و  $H_2$  به ترتیب با  $n$  و  $m$  عنصر داریم. درخت‌ها با اشاره گر پیاده سازی شده‌اند، یعنی هر گره به پدر و دو فرزند خود (در صورت وجود) اشاره گر دارد. الگوریتمی از مرتبه  $O(\log(n + m))$  ارائه کنید تا یک مین-هیپ جدید شامل همه  $n + m$  عنصر ایجاد نماید.

ب: دو درخت جستجوی دودویی با عناصر متفاوت، یکی با  $n$  و دیگری با  $m$  عنصر داریم. الگوریتمی از مرتبه  $O(n + m)$  ارائه کنید که یک درخت جستجوی دودویی جدید شامل همه عناصر دو درخت قبلی بسازد.

سوال شش: از سه سوال زیر به دو مورد پاسخ دهید (۳ نمره) ۲۰ دقیقه

الف: فرض کنید صد کلید یک تا صد را به یک درخت جستجوی دودویی اضافه کردیم. ترتیب اضافه شدن کلیدها معلوم نیست اما احتمال تمام حالات ترتیب‌های اضافه شدن کلیدها باهم برابر است. در صورتی که اولین کلید اضافه شده ۴۳ باشد، با چه احتمالی در روند اضافه شدن کلیدها به درخت، کلیدهای ۴ و ۹ باهم مقایسه می‌شوند؟

ب: کدام گزینه پیمایش پیشوندی یک درخت جست و جوی دودویی با پیمایش پسوندی به صورت زیر است؟

*Postorder:* 5, 6, 15, 10, 23, 24, 22, 26, 20

الف: 20, 23, 15, 5, 6, 10, 26, 24, 22

6, 5, 10, 15, 20, 22, 24, 23, 26      :ب

20,10,6,5,15,26,24,23,22 :ج

20, 10, 6, 5, 15, 26, 22, 24, 23      :D

ج: فرض کنید  $n$  عدد (نه لزوماً متمایز) بین  $0$  تا  $2^{100}$  داریم ولی از آنجایی که نمایش اعداد بزرگ سخت است، می‌خواهیم به هر کدام از این اعداد یک کلید از  $0$  تا  $n-1$  تخصیص دهیم به طوری که هر دو عدد یکسان که در مجموعه اولیه داشتیم یک کلید یکسان داشته باشند و از این به بعد برای نمایش آنها به جای خود عدد از این کلید کوچک جدید استفاده کنیم. الگوریتمی طراحی کنید که در  $O(n)$  این کار را انجام دهد. (راهنمایی: به ساختمان داده برای فکر کنید)

سوال هفت: درهم سازی (۵/۱ نمره) ۲۰ دقیقه

جدول T دارای ۱۰ خانه است. اگر از روش آدرس دهی باز و تابع جستجوی درجه دوم که به شکل  $h(k, i) = (h'(k) + i^2) \bmod m$  استفاده کنیم و داشته باشیم  $h'(k) = k \bmod 10$  و داده‌ها را به ترتیب زیر از چپ به راست وارد جدول کنیم، شکل نهایی جدول به چه صورتی خواهد بود؟ (توجه داشته باشید  $m = 10$  است)

18, 41, 22, 44, 59, 32, 31, 73

[illegible]