فصل

# ۲

# مرتب‌سازی

## ۱.۲ بله- خیر

**۱.۲** آیا روشی وجود دارد که بدون افزایش درجه‌ی پیچیدگی، بتوان هر الگوریتم مرتب‌سازی ناپایدار را به گونه‌ی پایدار آن تبدیل کرد؟

☐ بله ☐ خیر

**۲.۲** دنباله‌ی $A[1..n]$ زیگزاگی (bitonic) است اگر عناصر دنباله را به‌ترتیب دور یک دایره بنشانیم، می‌توان یک عنصر را یافت که دنباله‌ی عناصر بعدی ابتدا صعودی و سپس نزولی باشند. آیا یک دنباله‌ی زیگزاگی را می‌توان در $\mathcal{O}(n)$ مرتب کرد؟

☐ بله ☐ خیر

**۳.۲** آیا $n$ عدد صحیح و مثبت کوچک‌تر از $n^{100}$ را می‌توان در زمان خطی مرتب کرد؟

☐ بله ☐ خیر

**۴.۲** دنباله‌ای را در نظر بگیرید که در آن عناصر با اندیس زوج از هر دو عنصر مجاورش کوچک‌تر باشد، آیا می‌توان این‌چنین دنباله‌ای را در $\mathcal{O}(n)$ مرتب کرد؟

☐ بله ☐ خیر

**۵.۲** آیا یک درخت تصمیم به ارتفاع ۱۰ برای مرتب‌سازی ۷ عنصر وجود دارد؟

☐ بله ☐ خیر

۶.۲ آیا $n_0 > 1$ ای وجود دارد که برای هر $n > n_0$، آرایه‌ای با $n$ عنصر بتوان یافت که مرتب‌سازی درجی آن از مرتب‌سازی ادغامی سریع‌تر عمل کند؟

☐ بله ☐ خیر

۷.۲ آیا الگوریتم مرتب‌سازی مقایسه‌ای (comparison sort) وجود دارد که در هر آرایه‌ی $n$ تایی، تعداد عناصر مجزا را در زمان میانگین $\mathcal{O}(n)$ به‌دست بیاورد؟

☐ بله ☐ خیر

۸.۲ می‌خواهیم از بین $n$ عنصر، $\frac{n}{2}$ عنصر بزرگ‌تر را به‌ترتیب پیدا کنیم. آیا برای این‌کار الگوریتمی با زمان سریع‌تر از $\mathcal{O}(n \lg n)$ وجود دارد؟

☐ بله ☐ خیر

⋆ ۹.۲ فرض کنید ماشینی وجود دارد که برای هر $k$ ی دل‌خواه، $k$امین عنصر آرایه‌ی $n$ عنصری $A$ را در زمان $\mathcal{O}(\sqrt{n})$ محاسبه می‌کند. آیا با استفاده از این ماشین می‌توان آرایه‌ی $A$ را در زمان $\mathcal{O}(n)$ مرتب کرد؟

☐ بله ☐ خیر

۱۰.۲ آیا می‌توان الگوریتمی از مرتبه‌ی $\mathcal{O}(n)$ داشت که $n$ عدد موجود در یک هرم را به‌ترتیب در یک آرایه بنویسد؟

☐ بله ☐ خیر

⋆ ۱۱.۲ آیا به‌دست آوردن نزدیک‌ترین عنصر موجود در یک آرایه‌ی $n$ تایی به عدد ورودی $X$ توسط یک الگوریتم مقایسه‌ای از $\Omega(\lg n)$ است؟

$A[i]$ نزدیک‌ترین عنصر به $X$ است اگر هیچ $1 \leq j \leq n$ ای موجود نباشد که $|X - A[j]| < |X - A[i]|$.

☐ بله ☐ خیر

۱۲.۲ آرایه‌ای از $n$ عدد ۱، ۰، و ۱− داده شده است. آیا این آرایه را می‌توان در $\mathcal{O}(n)$ مرتب کرد؟

☐ بله ☐ خیر

⋆ ۱۳.۲ یک شبکه‌ی مرتب‌ساز با ۳ ورودی، برای ورودی‌های $\langle ۲, ۳, ۸ \rangle$، $\langle ۳, ۸, ۲ \rangle$، $\langle ۸, ۲, ۳ \rangle$ درست کار می‌کند. آیا این شبکه هر ورودی دیگر را هم به‌درستی مرتب می‌کند؟

☐ بله ☐ خیر

**۱۴.۲** الگوریتمی را در نظر بگیرید که آرایه‌ی $A[1..3n]$ از $3n$ عدد صحیح مجزا را دریافت می‌کند و دو عدد $x < y$ را تولید می‌کند به‌طوری‌که $n$ عنصر $A$ کم‌تر از $x$، $n$ عنصر دارای مقدار بین $x$ و $y$، و $n$ عنصر دارای مقدار بیش‌تر از $y$ هستند. چنین الگوریتمی از $\Omega(n \lg n)$ است.

☐ بله ☐ خیر

⋆ **۱۵.۲** الگوریتم تصادفی در زمان میانگین $\mathcal{O}(n)$ وجود دارد که تعیین کند آیا یک آرایه‌ی $n$ عنصری $A$ حاوی عناصر تکراری هست یا خیر.

☐ بله ☐ خیر

**۱۶.۲** عنصر بیشینه‌ی یک هرم کمینه با $n$ عنصر را می‌توان در $\mathcal{O}(\lg n)$ به‌دست آورد.

☐ بله ☐ خیر

⋆ **۱۷.۲** برای هر $\epsilon > 0$ دستِ‌کم $\Omega(n \lg n)$ زمان طول می‌کشد تا با دریافت $n$ عدد مجزا، تعداد $n^{1-\epsilon}$ عدد بیشینه‌ی این اعداد را به‌صورت مرتب به‌دست آورد.

☐ بله ☐ خیر

**۱۸.۲** $n$ نقطه‌ی $\langle x_1, y_1\rangle$، $\langle x_2, y_2\rangle$، ... تا $\langle x_n, y_n\rangle$ داده شده‌اند. فرض کنید که همه‌ی مختصه‌های $x_i$ و $y_i$ مجزا هستند. با فرض زوج بودن $n$، می‌خواهیم یک خط جداکننده‌ی عمودی $L$ بیابیم به‌طوری‌که تعداد نقاط سمت چپ این خط با تعداد نقاط سمت راست آن برابر باشد. این خط را می‌توان در $\mathcal{O}(n)$ به‌دست آورد.

☐ بله ☐ خیر

⋆ **۱۹.۲** آرایه‌ی $A[1..n]$ که $n = 2^k - 1$ حاوی تمام رشته‌های $k$ بیتی جز یک رشته‌ی نامشخص $x$ است. تنها عمل مجاز «خواندن $j$امین بیت از رشته‌ی $A[i]$» است که در زمان ثابت انجام می‌شود. آیا می‌توان $x$ را در زمان $\mathcal{O}(n)$ یافت؟

☐ بله ☐ خیر

⋆ **۲۰.۲** در مسئله‌ی پیشین (مسئله‌ی ۱۹.۲) اگر خواندن یک بیت $j$ از یک عدد $A[i]$ در زمان ثابت نیز مجاز باشد، آیا می‌توان عنصر مفقود را در زمان $\mathcal{O}(nk)$ و با حافظه‌ی $\mathcal{O}(k^2)$ بیت به‌دست آورد؟ دقت کنید که نگه‌داری تمام اعداد در حافظه به $\mathcal{O}(nk)$ بیت حافظه نیاز دارد.

☐ بله ☐ خیر

**۲۱.۲** مجموع کوچک‌ترین $\sqrt{n}$ عنصر از بین $n$ عنصر مجزا را می‌توان در زمان $\mathcal{O}(n)$ به‌دست آورد.

☐ بله ☐ خیر

**۲۲.۲** مرتب‌سازی سریع غیر تصادفی، یک آرایه‌ی $n$ عضوی مرتب را در زمان $\mathcal{O}(n \lg n)$ مرتب می‌کند.

☐ بله ☐ خیر

**۲۳.۲** یک گونه از الگوریتم مرتب‌سازی سریع را در نظر بگیرید که فقط بخش‌بندی‌ای را می‌پذیرد که متوازن باشد و در آن اختلاف تعداد عناصر دو بخش حداکثر ۳ باشد. زمان اجرای این الگوریتم مرتب‌سازی به‌طور میانگین $\mathcal{O}(n \lg n)$ است.

☐ بله ☐ خیر

**۲۴.۲** یک ورودی از $n$ وجود دارد که الگوریتم مرتب‌سازی سریع تصادفی همیشه در زمان $\mathcal{O}(n^۲)$ آن ورودی را مرتب می‌کند.

☐ بله ☐ خیر

## ۲.۲ مرتب‌سازی سریع

**۲۵.۲** احتمال آن‌که الگوریتم مرتب‌سازی سریع تصادفی $n$ عنصر را در زمان $\Omega(n^۲)$ مرتب کند دستِ‌کم برابر است با:

[۱] $۱/n$ [۲] $۱/(n!)$ [۳] $n/(n!)$ [۴] $n^۲/(n!)$

**۲۶.۲** اگر در مرتب‌سازی سریع محور همواره عنصر میانه باشد، در آن‌صورت پیچیدگی الگوریتم چگونه خواهد بود؟

[۱] همیشه $\mathcal{O}(n \log n)$
[۲] همیشه $\mathcal{O}(n^۲)$
[۳] در بدترین حالت $\mathcal{O}(n^۲)$ و در حالت میانگین برابر $\mathcal{O}(n \log n)$
[۴] در بدترین حالت $\mathcal{O}(n^۲ \lg n)$ و در حالت میانگین برابر $\mathcal{O}(n^۲)$

**۲۷.۲** در گونه‌ی جدید از مرتب‌سازی سریع، $۲\sqrt{n} + ۱$ عنصر اولِ آرایه‌ی $n$ عضوی $A$ را انتخاب می‌کنیم. سپس با یک الگوریتم ساده مانند مرتب‌سازی درجی این عناصر را مرتب می‌کنیم و میانه‌ی آن‌ها را به‌دست می‌آوریم. این میانه را محور قرار می‌دهیم و بقیه‌ی الگوریتم مرتب‌سازی را اجرا می‌کنیم. کدام‌یک از رابطه‌های زیر زمان اجرای این الگوریتم را در بدترین حالت نشان می‌دهد؟ (ارشد ۱۳۸۴)

[۱] $T(n) \leq T(n - \sqrt{n}) + \mathcal{O}(\sqrt{n})$
[۲] $T(n) \leq ۲T(n - \sqrt{n}) + \mathcal{O}(n)$
[۳] $T(n) \leq T(\sqrt{n}) + T(n - \sqrt{n}) + \mathcal{O}(n)$
[۴] $T(n) \leq T(۲\sqrt{n}) + T(n - ۲\sqrt{n}) + \mathcal{O}(n)$

**۲۸.۲** بدترین زمان اجرای الگوریتم مسئله‌ی پیشین (مسئله‌ی ۲۷.۲) کدام است؟

۱ $\mathcal{O}(n\lg n)$ ۲ $\mathcal{O}(n\sqrt{n})$ ۳ $\mathcal{O}(n\lg n\sqrt{n})$ ۴ $\mathcal{O}(n^{۲})$

**۲۹.۲** فرض کنید که رویه‌ی بخش‌بندی در الگوریتم مرتب‌سازی سریع به‌صورت روبه‌روست. اگر همه‌ی درایه‌های $A[p..r]$ دارای مقدارهای یکسانی باشند، مقداری که این رویه باز می‌گرداند، برابر با کدام‌یک از گزینه‌های زیر است؟ (ارشد ۱۳۸۱)

```
PARTITION (A, p, r)
  x ← A[p]; i ← p − 1; j ← r + 1
  while true
     do repeat j ← j − 1
          until A[j] ≤ x
          repeat i ← i + 1
          until A[i] ≥ x
          if (i < j)
            then SWAP(A[i], A[j])
            else    return j
```

۱ $p$ ۲ $p+۱$ ۳ $\lfloor \frac{p+r}{۲} \rfloor$ ۴ $r$

**۳۰.۲** کدام‌یک از گزینه‌های زیر در مورد مرتب‌سازی سریع با بیان متداول آن (انتخاب عنصر اول به‌عنوان محور) درست است؟

۱ اگر آرایه‌ی ورودی از قبل مرتب باشد، زمان اجرای آن $\Theta(n)$ است.
۲ اگر آرایه‌ی ورودی از قبل مرتب باشد، زمان اجرای آن $\Theta(n\log n)$ است.
۳ اگر آرایه‌ی ورودی برعکس مرتب شده باشد، زمان اجرای آن $\Theta(n\log n)$ است.
۴ اگر درایه‌های آرایه‌ی ورودی دارای مقدار یکسان باشند، زمان اجرای آن $\Theta(n\log n)$ است.

## ۳.۲ مرتبه‌ی آماری

⋆ **۳۱.۲** الگوریتم زیر از میان $n$ عنصر داده‌شده دو عنصر کمینه و بیشینه را پیدا می‌کند.

**۱.** عناصر را به تعدادی زوج عنصر به‌اضافه‌ی حداکثر یک عنصر تنها تقسیم کن.
**۲.** با یک مقایسه بین عناصر هر زوج، عنصر بزرگ‌تر و کوچک‌تر آن‌دو را پیدا کن.
**۳.** بین عناصر بزرگ‌تر عنصر بیشینه را پیدا کن ($max$).
**۴.** بین عناصر کوچک‌تر عنصر کمینه را پیدا کن ($min$).
**۵.** عنصر تنها را هم (در صورت وجود) با $max$ و $min$ مقایسه کن و بزرگ‌ترین و کوچک‌ترین عناصر نهایی را پیدا کن.

تعداد مقایسه‌های این الگوریتم چند تاست؟ دقت کنید که در سطر سوم و چهارم، از روش‌های شناخته‌شده و مرسوم برای یافتن عنصر بیشینه و کمینه استفاده می‌شود.

۱) $\lceil \frac{۳n}{۲} \rceil - ۲$ ۲) $\lceil \frac{۳n}{۲} - ۱ \rceil$ ۳) $\lceil \frac{۲n}{۳} \rceil - ۲$ ۴) $\lceil \frac{۲n}{۳} - ۱ \rceil$

⋆ **۳۲.۲** فرض کنید $X[۱ \cdots n]$ و $Y[۱ \cdots n]$ دو آرایه از اعداد دوبه‌دو متمایز و هر کدام به‌صورت غیرنزولی مرتب باشند. سریع‌ترین الگوریتم برای یافتن میانه‌ی تمامی این $۲n$ عدد از چه زمانی است؟

۱) $\mathcal{O}(۱)$ ۲) $\mathcal{O}(\lg n)$ ۳) $\mathcal{O}(\lg^{۲} n)$ ۴) $\mathcal{O}(\sqrt{n})$

**۳۳.۲** فرض کنید $n = ۲^k$. می‌خواهیم عنصر بیشینه را در یک ماتریس $A$ به‌اندازه‌ی $n \times n$ بیابیم. برای این‌کار، $A$ را به چهار ماتریس به‌اندازه‌ی $\frac{n}{۲} \times \frac{n}{۲}$ تقسیم می‌کنیم. بیشینه‌ی هر ماتریس را به‌صورت بازگشتی به‌دست می‌آوریم، بیشینه‌ی این بیشینه‌ها جواب است. تعداد دقیق مقایسه‌های عناصر $A$ با هم چند تاست؟ (ارشد ۱۳۸۹)

۱) $n^{۲} - ۱$ ۲) $۲n - ۱$ ۳) $۳ \lg n$ ۴) $۳(\lg n - ۱)$

**۳۴.۲** الگوریتم روبه‌رو دو عنصر بیشینه و کمینه‌ی یک آرایه‌ی $n$ عضوی $A$ را پیدا می‌کند. بیشینه و کمینه‌ی تعداد مقایسه‌های دو عنصر از درایه‌های مختلف $A$ به‌ترتیب برابر با کدام گزینه است؟

```
MINMAX-1 (A)
min ← 1; max ← 1
for i ← 2 to N
  do if A[i] < A[min]
       then min ← i
       else  if A[i] > A[max]
               then max ← i
```

۱) $n - ۱$ و $n - ۱$ ۲) $n$ و $n$
۳) $۲(n - ۱)$ و $n - ۱$ ۴) $۲n$ و $n$

**۳۵.۲** در الگوریتم یافتن عنصر میانه از $n$ عنصر، ما ابتدا عناصر را به دسته‌های ۵ تایی تقسیم می‌کنیم. میانه‌ی هر دسته را می‌یابیم و سپس میانه‌ی میانه‌ها را به‌طور بازگشتی پیدا می‌کنیم. از این عنصر به‌عنوان محور در بخش‌بندی کل آرایه استفاده می‌کنیم و پس از آن به‌صورت بازگشتی همین الگوریتم را بر روی یکی از بخش‌ها اجرا می‌کنیم. رابطه‌ی بازگشتی برای زمان اجرای این الگوریتم کدام است؟ (ارشد ۱۳۸۲)

۱) $T(n) \le T(\lfloor \frac{n}{۵} \rfloor) + T(\frac{۷n}{۱۰} + ۶) + \mathcal{O}(n)$
۲) $T(n) \le T(\lfloor \frac{n}{۵} \rfloor) + T(\frac{۳n}{۱۰} - ۶) + \mathcal{O}(n)$
۳) $T(n) \le T(\lceil \frac{n}{۵} \rceil) + T(\frac{۷n}{۱۰} + ۶) + \mathcal{O}(n)$
۴) $T(n) \le T(\lceil \frac{n}{۵} \rceil) + T(\frac{۳n}{۱۰} - ۶) + \mathcal{O}(n)$