دانشگاه تهران
دانشکدگان فارابی
دانشکده مهندسی

**طراحی و تحلیل الگوریتم‌ها**
**نیمسال دوم ۰۴–۱۴۰۳**
http://courses.fouladi.ir/algorithm

# تکلیف شماره‌ی ۲

## بخش دوم

# روش تقسیم و غلبه

Divide and Conquer Method

### ◊ مسئله‌های چندگزینه‌ای

**۱)** در جستجوی دودویی، زمان اجرای بدترین حالت (بر حسب تعداد مقایسه‌ها) با معادله‌ی بازگشتی $W(n) = W(\lfloor \frac{n}{۲} \rfloor) + ۱$ داده می‌شود. کدام گزینه حل این معادله‌ی بازگشتی است؟

۱) $W(n) = \lceil \log_۲ n \rceil$ ۲) $W(n) = \lfloor \log_۲ n \rfloor$
۳) $W(n) = \lfloor \log_۲ n \rfloor + ۱$ ۴) $W(n) = \lceil \log_۲ n \rceil + ۱$

**۲)** حل یک مسئله با استفاده از یک الگوریتم تقسیم و غلبه انجام شده است. در این الگوریتم، هر مسئله‌ی بزرگ به پنج مسئله‌ی کوچک‌تر شکسته شده و اندازه‌ی هر مسئله‌ی شکسته‌شده، یک سوم مسئله‌ی بزرگ‌تر می‌باشد. الگوریتم برای شکستن و ترکیب، نیاز به زمان $O(n)$ دارد. مرتبه‌ی اجرای این الگوریتم چیست؟

۱) $\Theta(n \log_۳ ۵)$ ۲) $\Theta(n^۳)$ ۳) $\Theta(n^{\log_۳ ۵})$ ۴) $\Theta(n^{۵/۳})$

**۳)** الگوریتم زیر برای محاسبه‌ی بزرگ‌ترین مقسوم علیه مشترک دو عدد صحیح نامنفی نوشته شده است.

```
GCD(a, b)
  if b = 0 then
  |   return a
  else
  |_  return GCD(b, a mod b))
```

اگر $a > b$ باشد، در این صورت می‌توان گفت که مرتبه‌ی زمانی این الگوریتم ... است.

۱) $O(\log_۲ a)$ ۲) $O(a \log_۲ b)$ ۳) $O(b \log_۲ a)$ ۴) $O(a/۲)$

**۴)** در گونه‌ی متفاوتی از مرتب‌سازی سریع (Quick-Sort)، برای انتخاب محور از میان $n$ عنصر، $۲\sqrt{n} + ۱$ عنصر اول آرایه را انتخاب می‌کنیم و با یک الگوریتم ساده مانند مرتب‌سازی درجی آنها را مرتب می‌کنیم؛ عنصر میانه‌ی این تعداد عنصر مرتب را به عنوان محور ( pivotالگوریتم) انتخاب می‌کنیم. بقیه‌ی الگوریتم مانند الگوریتم استاندارد عمل می‌کند. بدترین زمان اجرای این الگوریتم با کدام رابطه‌ی بازگشتی نشان داده می‌شود؟

۱) $T(n) = ۲T(\sqrt{n}) + n$ ۲) $T(n) = T(n - ۱) + n$
۳) $T(n) = ۲T(n - \sqrt{n}) + n$ ۴) $T(n) = T(n - \sqrt{n}) + T(\sqrt{n}) + n$

**۵)** در الگوریتم مرتب‌سازی ادغامی (Merge-Sort)، اگر به جای آنکه هر بار لیست به دو قسمت مساوی تقسیم شود، به چهار قسمت مساوی تقسیم شود و در مرحله از ترکیب، این چهار لیست در یکدیگر ادغام شوند، پیچیدگی زمانی الگوریتم $T(n)$ چه خواهد شد؟

۱) $O(n^{۳/۴})$ ۲) $O(n \log_۲ n)$ ۳) $O(n^۲ \log_۲ n)$ ۴) $O(n^۲ \log_۴ n)$

۶) یک ماشین انتزاعی که در مبنای ۱۰ کار می‌کند، اعداد را به راحتی با یکدیگر جمع و تفریق می‌کند و ضرب اعداد در $۱۰^x$ را از طریق شیفت به راحتی انجام می‌دهد، اما در ضرب، تنها قادر است اعداد یک رقمی را در هم ضرب نماید. برای ضرب دو عدد ۱۲۳۴ در ۵۶۱۸ حداقل تعداد عملیات ضرب ممکن چه تعدادی می‌باشد؟ (راهنمایی: از ضرب اعداد صحیح بزرگ استفاده کنید.)

۱) ۸ ۲) ۱۲ ۳) ۱۳ ۴) ۱۶

۷) برای حل یک مسئله با ورودی $n$ چند الگوریتم با مرتبه‌های زمانی مختلف پیشنهاد شده است. اگر زمان اجرا $T(n)$ باشد، کدام گزینه برای حل این مسئله با روش تقسیم و غلبه مناسب است؟

۱) $T(n) = T(n-۱) + T(n-۲) + ۱$ ۲) $T(n) = ۲T(n-۲) + ۱$
۳) $T(n) = T(n-۲) + ۲$ ۴) $T(n) = nT\left(\frac{n}{۱۰}\right) + ۱$

۸) الگوریتم ادغام Merge برای ادغام دو آرایه با طول‌های $n$ و $m$ به چند مقایسه و چند جابجایی نیاز دارد؟

۱) حداکثر $n+m$ مقایسه و دقیقاً $n+m$ جابجایی
۲) حداکثر $n+m-۱$ مقایسه و دقیقاً $n+m+۱$ جابجایی
۳) حداکثر $n+m-۱$ مقایسه و دقیقاً $n+m$ جابجایی
۴) حداکثر $n+m+۱$ مقایسه و دقیقاً $n+m-۱$ جابجایی

۹) الگوریتم افراز (Partition) در الگوریتم مرتب‌سازی سریع (Quick-Sort) به صورت زیر است:

```
Partition(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;
```

اگر تمام درایه‌های $A[p..r]$ دارای مقدار یکسانی باشند، مقداری که روال فوق برمی‌گرداند، چیست؟

۱) $p$ ۲) $r$ ۳) $\left\lceil \frac{p+r}{۲} \right\rceil$ ۴) $\left\lfloor \frac{p+r}{۲} \right\rfloor$

۱۰) الگوریتم افراز (Partition) برای افراز یک آرایه‌ی $n$تایی، حداکثر به چند مقایسه‌ی کلید نیاز دارد؟

۱) $n$ ۲) $n-۱$ ۳) $n+۱$ ۴) $\lfloor n/۲ \rfloor$

۱۱) در الگوریتم یافتن $k$امین بزرگ‌ترین عنصر از میان $n$ عنصر، ابتدا همه‌ی عناصر را به دسته‌های ۵تایی (به‌جز احتمالاً دسته‌ی آخر) تقسیم می‌کنیم. میانه‌ی هر دسته را به‌دست می‌آوریم و سپس میانه‌ی میانه‌ها را به صورت بازگشتی پیدا می‌کنیم. این عنصر را به‌عنوان محور (pivot) انتخاب می‌کنیم و الگوریتم افراز (Partition) را بر روی عناصر آرایه اجرا می‌کنیم. پس از آن، همین الگوریتم را به صورت بازگشتی (و برای یک $k$ دیگر) بر روی یکی از بخش‌ها اجرا می‌کنیم تا عنصر مورد نظر پیدا شود. زمان اجرای این الگوریتم توسط کدام یک از رابطه‌های بازگشتی زیر بیان می‌شود؟

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

۱۲) اگر الگوریتم جستجوی دودویی را برای همه‌ی عناصر آرایه‌ی $A[۱..۸] = [۵, ۱۰, ۱۵, ۲۰, ۲۵, ۳۰, ۳۵, ۴۰]$ به‌کار ببریم، میانگین تعداد مقایسه‌ها برای جستجوی موفق تقریباً کدام است؟

۱) ۲٫۲ ۲) ۲٫۴ ۳) ۲٫۶ ۴) ۲٫۸


Prepared by Kazim Fouladi <kazim@fouladi.ir>, Spring 2025 (۹ اسفند ۱۴۰۳)

## ◊ مسئله‌های تشریحی

۱) الگوریتم زیر، بیان یک الگوریتم مرتب‌سازی است که توسط پروفسور هوارد مطرح شده است:

**ابتدا دو عنصر اول و آخر آرایه را با هم مقایسه می‌کنیم، اگر ترتیب این دو برعکس بود، جای آنها را عوض می‌کنیم. سپس آرایه را به سه قسمت مساوی (تقریبا) تقسیم می‌کنیم. آن‌گاه به صورت بازگشتی با همین روش زیرآرایه‌ی متشکل از قسمت اول و دوم آرایه را مرتب می‌کنیم، سپس زیرآرایه‌ی متشکل از قسمت دوم و سوم و آن‌گاه زیرآرایه‌ی متشکل از قسمت اول و دوم را دوباره مرتب می‌کنیم. مسئله وقتی به اندازه‌ی کافی کوچک است که زیرآرایه، دو عنصر داشته باشد.**

(الف) شبه کدی بنویسید که این الگوریتم را پیاده‌سازی کند.

(ب) زمان اجرای این الگوریتم را در همه‌ی موارد تعیین کنید.

۲) الگوریتمی ارائه دهید که در زمان $O(n \log_۲ n)$ تعداد وارونگی‌های (inversion) یک آرایه‌ی $n$تایی $A$ را بیابد.
(راهنمایی: می‌توانید الگوریتم Merge-Sort را تغییر دهید).
یادآوری می‌شود که در آرایه‌ی $n$تایی $A[۱..n]$ با عناصر متمایز، زوج $(i,j)$ یک وارونگی نام دارد اگر $i<j$ و $A[i]>A[j]$ باشد.


Prepared by Kazim Fouladi <kazim@fouladi.ir>, Spring 2025 (۹ اسفند ۱۴۰۳)