



به نام خدا

تمرین اول ساختمان داده‌ها و الگوریتم

الگوریتم‌ها و حل مسئله - پیچیدگی زمانی - الگوریتم‌های بازگشتی

استاد:

دکتر رستمی

استادیار:

عرشیا عموزاد

۱. توابع زیر را از نظر رشد با هم مقایسه کنید

$$g(n) = n^{\log n}$$
$$f(n) = (\log n)^n(n)$$

۲. کدام یک از گزاره‌های زیر دربارهٔ توابع ذکر شده صحیح است؟

$$f(n) = 4^{\log n}$$
$$g(n) = (\log n)^{\log n}$$
$$h(n) = \log^2 n$$

- $f(n) = O(g(n), g(n) = \Omega(h(n)))$
- $g(n) = \Omega(h(n), h(n) = \Omega(f(n)))$
- $f(n) = O(h(n), g(n) = \Omega(f(n)))$
- $h(n) = O(g(n), f(n) = \Theta(g(n)))$

۳. کد زیر الگوریتم جستجوی خطی را نشان می‌دهد
فرض کنید با احتمال ۵۰ درصد عنصر target در آرایه arr است. بطور متوسط چند عنصر برای یافتن target برسی می‌شوند؟

```
1 def linear_search(arr, target):
2     for index in range(len(arr)):
3         if arr[index] == target:
4             return index
5     return -1
```

۴. آیا می‌توان ادعا کرد که برای هر دوتابع $f(n) = O(g(n))$ یا $g(n) = O(f(n))$ یا هر دو؟

۵. توابع زیر را از نظر نرخ رشد به ترتیب صعودی مرتب کنید.

$$f_5(n) = n^{\log \log n}, f_4(n) = n^{\log n}, f_3(n) = (\log n)^{\log n}, f_2(n) = 2^{\sqrt{n}}, f_1(n) = n^{1.01}$$

۶. در یک زمستان سرد، خرس قطعه‌گوشت دغیقاً به اندازه‌های ۱، ۲ تا n را در غاری ذخیره کرده است. او هر روز یکی از این قطعه‌گوشت‌ها را به صورت تصادفی انتخاب می‌کند. اگر اندازه‌ی گوشت عدد فردی بود، آن را کاملاً می‌خورد. اگر زوج بود، آن را دقیقاً نصف می‌کند، یک نصف آن را می‌خورد و نصف دیگر را مجدداً در غار قرار می‌دهد. اگر گوشتی موجود نباشد خرس می‌میرد. با این الگوریتم، برای n ‌های خیلی بزرگ روزهای باقیمانده از عمر خرس ما تابع کدام یک از گزینه‌های خواهد بود؟

$$O(n^2) \quad O(n \log n) \quad O(\log n) \quad O(n) \quad O(1)$$

۷. پیچیدگی زمانی توابع زیر را پیدا کنید.

$$T(n) = 4T\left(\frac{n}{2}\right) + \log^2 n! .1$$
$$T(n) = 3\sqrt[3]{n^2}T(\sqrt[3]{n}) + n \cdot \log n .2$$
$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{n}{6}\right) + n^{\sqrt{\log n}} .3$$
$$H(0) = 0, H(n) = \frac{1}{2-H(n-1)} .4$$
$$f(n) = \log(n!) .5$$

۸. الگوریتم زیر $(\binom{n}{m})$ را برای اعداد صحیح $n \geq m$ محاسبه می‌کند:

```
1 function combination(n, m:integer): integer;
2 begin
3     if (m = n) or (m = 0) then
4         return 1
5     else
6         return combination(n-1, m) + combination(n-1, m-1)
7     end;
```

برای $n > m$ دلخواه عمل جمع چند بار تکرار می‌شود؟
 $n(n-m) \bigcirc \binom{n}{m} - 1 \bigcirc nm \bigcirc \binom{n}{m} \bigcirc$

۹. درباره‌ی اجرای الگوریتم مرتب سازی ادغامی (Merge Sort) بر روی آرایه‌ای به طول n به سوالات زیر پاسخ دهید.

۱. اگر در مرحله‌ی تقسیم، الگوریتم به جای نصف شدن دقیق به نسبت های دلخواه مانند $\frac{1}{3}$ و $\frac{2}{3}$ یا $\frac{1}{4}$ و $\frac{3}{4}$ تقسیم شود پیچیدگی زمانی الگوریتم چگونه تغییر می‌کند؟
۲. فرض کنید همه‌ی عناصر آرایه از قبل بصورت صعودی مرتب شده باشند. در این صورت تعداد مقایسه‌های الگوریتم نسبت به حالت تصادفی چه تغییری می‌کند؟
۳. با اجرای برنامه‌ی زیر چه عددی چاپ می‌شود؟

```
1 #include <stdio.h>
2
3 int func(int, int);
4
5 void main() {
6     int n = 2;
7     int m = 3;
8     printf("%d\n", func(n, m));
9 }
10
11 int func(int m, int n) {
12     if (m == 0) return n + 1;
13     if (n == 0) return func(m - 1, 1);
14     return func(m - 1, func(m, n - 1));
15 }
```