

# تمرین سری اول درس طراحی الگوریتم

دانشگاه کاشان - استاد جواد سلیمی

دستیاران آموزشی: محمدعرفان هومانفر، محمدخورشا

- برای سوالات راه حل و توضیحات کامل ارائه کنید. و به صورت پی دی اف یا ورد در قالب ارایه شده بارگذاری کنید
- سوالاتی که تحت عنوان کد نویسی هستند به یکی از زبان های python, C++, C نوشته شود و فایل اجرایی آن ها ارسال شود.

۱- دو الگوریتم با پیچیدگی های زمانی  $T_1 = O(n \log n)$  و  $T_2(n) = O(n^2)$  داده شده اند. در چه بازه هایی از  $n$  به عنوان ورودی  $T_1(n)$  کارآمدتر از  $T_2(n)$  است. با تحلیل نشان دهید که کارآمدتر است.

۲- پیچیدگی زمانی تابع های زیر را تحلیل کنید و  $O$ -big آن ها را بدست آورید.

```
def recursive_function(n):  
    if n <= 1:  
        return 1  
    else:  
        return recursive_function(n - 1) + recursive_function(n - 1)
```

```
for (int t=1; t<=n; t=t*2){  
    for (int i=1; i<=n; i++){  
        x=x+1  
    }  
}
```

```
for (int i=1; i<=n*n*n; i++){  
    for (int i=1; i<=n; i++){  
        x=x+1;  
    }  
}
```

```
int x=0;  
int i=1;  
while (i<n){  
    x++;  
    i=i*i*i;  
}
```

۳- در درس ساختمان داده با insertion sort و binary search آشنا شده اید، کد مربوط به binary insertion sort را پیاده سازی کنید، این مدل از insertion sort بجای پیمایش پشت سر هم موارد برای پیدا کردن مکان insert آیتم از binary search برای این کار استفاده میکند.

- بعد از پیاده سازی تحلیل الگوریتم insertion sort را با binary insertion sort مقایسه کنید

۴- درستی یا نادرستی گزاره های زیر را بررسی کنید و یا شرایط درستی را بیان کنید

- (a) اگر  $f(n) \in O(g(n))$  آنگاه  $2^{f(n)} \in O(2^{g(n)})$
- (b) اگر  $f(n) \in o(g(n))$  آنگاه  $2^{f(n)} \in O(2^{g(n)})$
- (c) اگر  $f(n) \in o(g(n))$  آنگاه  $\log g(n) \in O(\log g(n))$
- (d)  $f(n) \in O(f^2(n))$
- (e)  $f(n) \in \theta\left(f\left(\frac{n}{2}\right)\right)$
- (f)  $f(n) + O(f(n)) \in \theta(f(n))$

۵- براساس ترتیب رشد مقایسه کنید.

$$\log^2 n - 2^{\log n} - n^3 - n^{\frac{1}{\log n}} - \ln \ln n - e^n - \ln n - n!$$

$$- 2^{\sqrt{2 \log n}} - \sqrt{2}^{\log n} - n^2 - 2^n - 2^{2^n} - \sqrt{\log n}$$

