تمرین سری اول

- برای سوالات راه حل و توضیحات کامل ارائه کنید.
- سوالاتی که تحت عنوان کد نویسی هستند به یکی از زبان های C, C++, python نوشته شود و فایل اجرایی آن ها را ارسال کنید.

۱- دو الگوریتم با پیچیدگی‌های زمانی $T_1 = O(nlogn)$ و $T_2(n) = O(n^2)$ داده شده‌اند. اندازه ورودی n که در آن $T_1(n)$ کارآمدتر از $T_2(n)$ می‌شود را تعیین کنید. با تحلیل نشان دهید که کارآمدتر است

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

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

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

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

d
```java
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 $2^{f(n)} \in O\left(2^{g(n)}\right)$

b) اگر $f(n) \in o(g(n))$ آنگاه $2^{f(n)} \in O\left(2^{g(n)}\right)$

c) اگر $f(n) \in o(g(n))$ آنگاه $\log g(n) \in O(\log g(n))$

d) $f(n) \in O\left(f^2(n)\right)$

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.

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

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