

تمرین سری اول

فصل اول

کارایی تحلیل و مرتبه الگوریتم ها

(۱) درستی و نادرستی عبارات را نشان دهید:

- a) $f^2(n) \in \Omega(f(n))$
- b) $1 + 2 + 3 + \dots + n \in \theta(n^2)$
- c) $f(x) \in \theta(g(x))$ اگر و تنها اگر $f(x) \in O(g(x))$ و $f(x) \in O(g(x))$
- d) $n \in O(2^n)$ و $\log n \in O(n)$

(۲) برای الگوریتم زیر:

```
int add_them(int n, int A[])
{
    index i,j,k;
    j=0
    for(i=1;i<=n;i++)
        j=j+A[i];
    k=1;
    for(i=1;i<=n;i++)
        k=k+k;
    return j+k;
}
```

- (a) اگر $n=5$ و آرایه شامل ۲ و ۳ و ۵ و ۷ و ۸ باشد. خروجی چیست؟
- (b) پیچیدگی الگوریتم را به دست آورید.
- (c) آیا راهی برای ساده سازی الگوریتم وجود دارد؟ سعی کنید کد را بهبود ببخشید.

(۳) پیچیدگی زمانی کدهای زیر را محاسبه کنید.

a)

```
for(int t=1 ; t<=n-1 ;t++){  
    for(int i=1 ; i<=n-t ; i++){  
        j=i+t;  
        for(int k=i ; k<=j-1 ;k++){  
            x=x+1;  
        }  
    }  
}
```

b)

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

c)

```
for(int i=1 ; i<=n3 ;i++){  
    for(int j=1 ; i<=4i ; j++){  
        for(int k=1 ; k<=3n ;k++){  
            x=x+1  
        }  
    }  
}
```

d)

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

e)

```
int x=0;  
int i=2;  
while(i<n){  
    x++;  
    i=i*i*i;  
}
```

f)

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

۴) الگوریتم insertion sort را به گونه ای تغییر دهید که برای پیدا کردن موقعیتی که باید insertion sort بعدی رخ دهد از الگوریتم جستجوی دودویی استفاده کند.

پیچیدگی این الگوریتم را در حالت worst case , best case, average case محاسبه کنید و با الگوریتم تغییرنیافته insertion sort مقایسه کنید.

- ۵) فرض کنید شما با یک دنباله از داده‌های تصادفی مواجه هستید و می‌خواهید یک زیرسری از آن‌ها پیدا کنید که شبیه به سری فیبوناچی باشد.
- الف) الگوریتمی طراحی کنید که بتواند وجود یک زیرسری فیبوناچی (یا شبیه به آن) را در داده‌های تصادفی پیدا کند.
- ب) پیچیدگی زمانی الگوریتم را تحلیل کنید.
- ج) آیا می‌توانید الگوریتم را به گونه‌ای بهینه‌سازی کنید که کارایی آن برای داده‌های بسیار بزرگ افزایش یابد؟