

سوال ۱

الف) رشد تقریبی توابع زیر را با هم مقایسه کنید.

$$f(n) \quad g(n)$$

$$a) \quad n^{\log n} \quad (\log n)^n$$

$$b) \quad \frac{n^2}{\log n} \quad n(\log n)^2$$

ب) درستی یا نادرستی گزاره‌های زیر را اثبات کنید. (برای اثبات نادرستی کفایت یک تابع مثال نقض بزنید.)

$$۱. f(n) + g(n) = \Theta(\max(f(n), g(n)))$$

$$۲. \text{ If } f(n) = O(g(n)) , \text{ then } h(f(n)) = O(h(g(n)))$$

$$۳. f(n) + O(f(n)) = \Theta(f(n))$$

$$۴. f(n) = \Theta(f(\frac{n}{2}))$$

سوال ۲

عبارات زیر را اثبات کنید.

(الف)

$$T(n) = n + \frac{n}{2} + \frac{n}{2} + \frac{n}{4} + \frac{n}{4} + \frac{n}{4} + \frac{n}{4} + \dots + \frac{n}{2^{\lceil \log_2(n) \rceil}} \implies T(n) \in O(n \log n)$$

(ب)

$$T(n) = n + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} + \frac{n}{5} + \dots + \frac{n}{n} \implies T(n) \in O(n \log n)$$

(راهنمایی: از الف استفاده کنید.)

(ج)

$$T(n) = 1 + 2 + \dots + n \implies T(n) \in \Theta(n^2)$$

(د)

اگر

$$a_1 + a_2 + \dots + a_m = n,$$

$$1 \leq a_1 < a_2 < \dots < a_m$$

آنگاه

$$m \in O(\sqrt{n})$$

(راهنمایی: از ج استفاده کنید.)

سوال ۳

با توجه به شبه‌کدها به موارد زیر پاسخ دهید.

(الف)

```
1. for(i=1 ; i<=n ; i++)
2.   for(j=2*i ; j<=n ; j+=i)
3.     is_Prime[j] = false
```

ثابت کنید خط ۳ از $O(n \log n)$ بار اجرا می‌شود.

(ب)

```
1. for(i=1 ; i<=n ; i++)
2.   a = random number between 1 to i
3.   if(a == 1)
4.     x = x + 1
```

ثابت کنید در *average case* مقدار x ، $O(\log n)$ است.

(ج) امتیازی

```
1. for(i=2 ; i<=n ; i++)
2.   j = i
3.   while(j<=n)
4.     j = j*j
```

۱. ثابت کنید خط ۳ از $O(n \log n)$ بار اجرا می‌شود.

۲. ثابت کنید همین خط از $O(n \log(\log n))$ بار اجرا می‌شود.

سوال ۴

شبه کد زیر یک نمونه‌ی تغییر کرده مرتب‌سازی درجی است. این کد آرایه را با شروع از `start` (به جای ۰ در حالت معمولی) مرتب می‌کند. برای مثال آرایه زیر:

[5, 2, 1, 3, 4]

با فرض `start = 2` پس از مرتب‌سازی تبدیل به آرایه زیر می‌شود:

[4, 5, 1, 2, 3]

یا در این آرایه:

[7, 8, 4, 2, 5, 3, 9, 6]

با فرض `start = 4` پس از مرتب‌سازی تبدیل به آرایه زیر می‌شود:

[6, 7, 8, 9, 2, 3, 4, 5]

با توجه به شبه‌کد زیر به موارد زیر پاسخ دهید.

```
Insertion_Sort(A[0,1,...,n-1])
start = a number from 0 to n-1
for(i=(start+1) mod n ; i!=start ; i=(i+1) mod n)
    key = A[i]
    j = (i+n-1) mod n
    while(j != (start+n-1) mod n and A[j] > key)
        A[(j+1) mod n] = A[j]
        j = (j+n-1) mod n
    A[(j+1) mod n] = key
```

الف) درستی الگوریتم را اثبات و پیچیدگی زمانی آن را محاسبه کنید.

ب) آرایه‌ای به اندازه n پیدا کنید که اگر $start = \lfloor \frac{n}{2} \rfloor$ باشد، الگوریتم کمترین تعداد عملیات‌های ممکن را برای مرتب سازی آرایه انجام می‌دهد.