بسمه تعالی

- تمرین سری اول درس ساختمان داده‌ها و مبانی الگوریتم‌ها
- پاسخ تمرین در قالب یک فایل pdf تایپ شده یا دست نویس اسکن شده (مرتب و خوانا) و با نام StudentNumber_HW1.pdf آپلود شود.
- مهلت ارسال تمرین تا ساعت 11:59 روز جمعه مورخ ۲۳ مهر ۱۴۰۰ می باشد.
- در صورتی که درمورد این تمرین سوال یا ابهامی داشتید با ایمیل Aut.dsfall1400@gmail.com با تدریس یاران در ارتباط باشید.

همچنین خواهشمند است در متن ایمیل به شماره دانشجویی خود اشاره کنید.

تحلیل الگوریتم

۱-

الف) پیچیدگی زمانی قطعه کد زیر را حساب کنید. (مراحل محاسبه را بنویسید)

```
int j = 0;

for (int i = 0; i < n; i++) {
        if (A[i] != B[j]) {
                int k = j;
                while (A[i] != B[k]) {
                        k++;
                }
                swap(B[j], B[k]);  // This function has constant time complexity
        }
        j++;
}
```

ب) (امتیازی) با توجه به قطعه کد بالا بگویید که الگوریتم ما چه کاری را دارد انجام میدهد؟

۲ - (امتیازی)

ثابت کنید در قطعه کد زیر در حالت میانگین، تعداد دفعاتی که عبارت Hello world چاپ می‌شود از مرتبه زمانی $O(log\ n)$ است.

```
for (int i = 0; i < n; i++) {
        randomNumber = A random number from 1 to i ();
        if (randomNumber == 1)
                print("Hello world");
}
```

مرتب سازی

۳-

اگر الگوریتم Bubble Sort را به جای شروع از خانه اول آرایه از خانه‌ای با index i شروع کنیم در این صورت آرایه ما بر اساس خانه i ام آرایه مرتب می شود.

این نوع مرتب سازی نیز یک روش مرتب سازی است که باعث می شود ترتیب اعضا در آرایه مرتب شده اما با یک shift همراه شود. به عنوان مثال اگر آرایه زیر را در نظر بگیریم:

3,4,2,1,5

اگر مرتب سازی بر اساس خانه دوم انجام شود خروجی زیر را میدهد:

4,5,1,2,3

و اگر از خانه چهارم انجام شود خروجی زیر را میدهد:

2,3,4,5,1

به این روش مرتب سازی I-index Bubble Sort میگوییم.

حال فرض کنید یک آرایه مرتب شده توسط I-index Bubble Sort دارید. می خواهیم یک مقدار مخصوص T را در آن جست و جو کنیم.

مثال:

4,5,6,1,2,3

T(3) $\longrightarrow$ 5

T(4) $\longrightarrow$ 0

الف) الگوریتمی طراحی کنید که با پیچیدگی زمانی $O(n)$ این جست و جو را انجام دهد. (الگوریتم را نوشته و طریقه محاسبه پیچیدگی زمانی آن را شرح دهید)

ب) حال سعی کنید الگوریتمی ارائه دهید که همین جست و جو را با پیچیدگی زمانی $O(log\ n)$ انجام دهد.

ج) ثابت کنید که الگوریتم شما در قسمت ب از اوردر $O(log\ n)$ می باشد.

۴ - قطعه کد زیر شبه کد مرتب سازی حبابی است.

```
Bubble_sort(A):
    for i ← n down to 1 do
        for j ← 1 to i do
            if A[j] > A[j+1]:
                swap(A[j], A[j+1])
            end if
```

الف) پیچیدگی زمانی قطعه کد فوق را در بهترین حالت، بدترین حالت و حالت میانگین محاسبه کنید.

ب) آیا امکان دارد با ایجاد تغییری در شبه کد فوق، پیچیدگی زمانی بهترین حالت بهبود یابد؟ در صورت پاسخ مثبت، آن تغییر را اعمال کنید.

رشد توابع

۵- موارد زیر را ثابت کنید.

نکته: زمانی که می‌خواهید ثابت کنید $A = \Theta(B)$ باید هر دو عبارت $A = O(B)$ و $A = \Omega(B)$ را اثبات کنید.

a) $n^{\frac{1}{\lg n}} = \Theta(1)$
b) $n! = \omega(2^n)$
c) $n! = o(n^n)$

۶- صحیح یا غلط بودن گزاره‌های زیر را اثبات کنید(در صورتی که گزاره ها غلط هستند مثال نقض کافیست)

a) $f(n) = O(g(n)) \Rightarrow g(n) = O(f(n))$
b) $f(n) = O(g(n)) \Rightarrow 2^{f(n)} = O(2^{g(n)})$

c) $f(n) = O(f(n^2))$

d) $f(n) = O(g(n)) \Rightarrow g(n) = \Omega(f(n))$

e) $f(n) = \Theta(f(\frac{n}{2}))$

f) $f(n) + o(f(n)) = \Theta(f(n))$

۷- مشخص کنید که به ازای هر جفت (A , B) آیا A از Θ, ω, Ω,o,O تابع B هست یا خیر(مانند ردیف اول)

A	B	O	o	Ω	ω	Θ
n^2	n^3	yes	yes	no	no	no
$lg^k n$	n^ϵ					
n^k	c^n					
2^n	$2^{n/2}$					
$2^{2lg\, n}$	n^2					
$n!$	$n.2^n$					
$n^{lg(lg(n))}$	$(lg(n))^{lg(n)}$					
$n^{lg(n)}$	$(log\, n)^2$					
$\frac{n^2}{log\, n}$	$n(log\, n)^2$					
$n^{\frac{1}{n}}$	$\sqrt{2}^{log\, n}$					