باسمه تعالی

-تمرین سری اول درس ساختمان داده ها و مبانی الگوریتم ها

-پاسخ تمرین در قالب یک فایل pdf تایپ شده یا دست نویس اسکن شده (مرتب و خوانا) و با نامHW1_StudentNumber.pdf آپلود شود.

-مهلت ارسال تمرین تا ساعت ۱۳:۰۰ روز دوشنبه مورخ ۱۳ اسفند ۱۳۹۷ می باشد.

- در صورت وجود هرگونه سوال می توانید با ایمیل های زیر در ارتباط باشید.

aliabigdeli@gmail.com amoazeni75@gmail.com

۱ : با در نظر گرفتن الگوریتم مرتب سازی حبابی(Bubblesort) به سوالات زیر پاسخ دهید :

```
BUBBLESORT(A)
1  for i ← 1 to length[A]
2      do for j ← length[A] downto i + 1
3             do if A[j] < A[j − 1]
4                   then exchange A[j] ↔ A[j − 1]
```

الف: فرض کنید A' خروجی مرتب شده ی الگوریتم را نشان دهد. برای اثبات درستی الگوریتم مرتب سازی حبابی نیاز است تا اثبات کنیم اولا الگوریتم پایان پذیر است، دوما شرط زیر برقرار است :

$$A'[1] \leq A'[2] \leq \cdots \leq A'[n]$$

که در آن n برابر با length[A] است، به نظر شما چه شرط دیگری برای اثبات درستی الگوریتم نیاز است؟

ب: برای حلقه موجود در الگوریتم(خط ۲ الی ۴) ثابت حلقه (Loop invariant) را به طور دقیق مشخص کنید و اثبات کنید که حلقه پایان پذیر است. برای مطالعه بیشتر در ارتباط با روش اثبات می توانید به کتاب مرجع درس مراجعه کنید.

ج : حال با توجه به ثابت حلقه ی تایین شده در قسمت قبل، ثابت حلقه خط ۱ الی ۴ را نیز به صورت دقیق مشخص کنید و در نهایت اثبات کنید که پس از پایان الگوریتم نامساوی $A'[1] \le A'[2] \le \cdots \le A'[n]$ برقرار خواهد بود.

برای مطالعه بیشتر در ارتباط با روش اثبات می توانید به کتاب مرجع درس مراجعه کنید.

د : پیچیدگی زمانی اجرای الگوریتم مرتب سازی حبابی در بدترین حالت چیست؟ الگوریتم مرتب سازی حبابی را با الگوریتم مرتب سازی درجی(Insertion sort) از نظر زمانی مقایسه کنید

۲ : یکی از روش های پیاده سازی مرتب سازی درجی به صورت بازگشتی می باشد، با در نظر گرفتن اینکه برای مرتب کردن دنباله A[1..n] ما ابتدا به صورت بازگشتی دنباله ی A[1.. n-1] را مرتب کرده و سپس A[n] را به دنباله ی مرتب شده اضافه می کنیم، تحلیل زمانی برای این نسخه ارائه دهید.

۳: مرتب سازی n عدد که در آرایه ی A ذخیره شده اند را با پیدا کردن اولین کوچکترین عنصر A و جابه جا کردن آن با عنصر A[1] در نظر بگیرید. سپس دومین کوچکترین عنصر را یافته و با A[2] جابه جا می کنیم. این روند را برای n-1 عنصر اول A انجام می دهیم.

الف : شبه کدی برای این الگوریتم مرتب سازی که به آن مرتب سازی انتخابی (Selection Sort) نیز می گویند بنویسید.

ب : ثابت حلقه ی این الگوریتم را مشخص کنید.

ج : چرا به جای آنکه برای n عنصر اول الگوریتم را اجرا کنیم برای n-1 عنصر اول الگوریتم را اجرا می کنیم؟

د: تحلیل زمانی برای بهترین حالت و بدترین حالت در قالب θ (Theta) ارائه دهید.

۴: برای هر یک از ردیف های جدول زیر مشخص کنید که رابطه ی بین A و B چگونه است (مثلا A یک big O، little o، Theta و ... برای B می باشد یا خیر).فرض کنید K>1 و 0 < ε و 1 < c است. کافی است در هر خانه از جدول Yes یا No بنویسید.

	A	B	O	o	Ω	ω	Θ
a.	$\lg^k n$	n^ϵ					
b.	n^k	c^n					
c.	$\sqrt{n}$	$n^{\sin n}$					
d.	2^n	$2^{n/2}$					
e.	$n^{\lg c}$	$c^{\lg n}$					
f.	$\lg(n!)$	$\lg(n^n)$					

۵ : فرض کنید $f(x)$ و $g(x)$ دو تابع مثبت باشند، برای درستی یا نادرستی هریک از موارد زیر اثبات ارائه دهید.

a. $f(n) = O(g(n))$ implies $g(n) = O(f(n))$.

b. $f(n) + g(n) = \Theta(\min(f(n), g(n)))$.

c. $f(n) = O(g(n))$ implies $\lg(f(n)) = O(\lg(g(n)))$, where $\lg(g(n)) \geq 1$ and $f(n) \geq 1$ for all sufficiently large n.

d. $f(n) = O(g(n))$ implies $2^{f(n)} = O(2^{g(n)})$.

e. $f(n) = O((f(n))^2)$.

f. $f(n) = O(g(n))$ implies $g(n) = \Omega(f(n))$.

g. $f(n) = \Theta(f(n/2))$.

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