

۱. به ازای هر یک از دو ترکیب عبارت A و B که در جدول زیر ذکر شده است مشخص کنید عبارت A از O , ω , Ω , نسبت به عبارت B است یا خیر. K و C دو عدد ثابت بزرگتر از یک هستند. استدلال خود به ازای هر سطر را بیان کنید.

A	B	O	O	Ω	ω	Θ
n^k	c^n					
\sqrt{n}	$n^{\sin(n)}$					
2^n	$2^{n/2}$					
$n^{\lg(c)}$	$c^{\lg(n)}$					
$\lg(n!)$	$\lg(n^n)$					
$\lg(n!)$	$(\lg(n))!$					

۲. هر یک از عبارات زیر را اثبات و یا با مثال نقض رد کنید.

- I. $f(n) = O(g(n))$ implies $g(n) = O(f(n))$
- II. $f(n) + g(n) = \Theta(\min(f(n), g(n)))$
- III. $f(n) = O(g(n))$ implies $g(n) = \Omega(f(n))$
- IV. $f(n) = \Theta(f(n / 2))$

۳. به طور خلاصه الگوریتم زیر را از نظر زمانی تحلیل کنید :

```
void sort(int *array, int size){
    for(int i = 0; i < size; i++){
        while(array[i] != i){
            swap(array[i], array[array[i]]);
        }
    }
}
```

ورودی این تابع یک جایگشت از اعداد ۰ تا $n-1$ است و خروجی این تابع یک آرایه ی مرتب شده است.

۴. در یک آرایه ی مرتب شده به طول n عنصری را بیابید که مقدار آن با شماره ی خانه ی آن برابر باشد.
مثال : در آرایه ی

$\{-1, 0, 3, 5\}$

سومین عنصر آرایه مقدار ۳ دارد (;
توجه شود که الگوریتم شما باید از $O(\log n)$ باشد.

۵. در یک دنباله ی اعداد، عدد غالب عددی است که بیش از $n/2$ بار تکرار شده باشد. الگوریتمی با $O(\log n)$ ارائه دهید که در یک دنباله اعداد مرتب شده به طول n در صورت وجود عدد غالب، آن عدد را پیدا کند.
نکته : عدد غالب در صورت وجود، یکتاست (;

۶. عنصر مینیمال : در یک آرایه ی یک بعدی از طول n خانه‌ای را مینیمال مینامیم که مقدار آن از دو خانه ی مجاور خود(در صورت وجود) کمتر باشد. الگوریتمی از $O(\log n)$ ارائه دهید که خانه ی مینیمال را در صورت وجود پیدا کند. در صورتی که بیش از یک خانه ی مینیمال وجود داشته باشد پیدا کردن یکی از آن‌ها کافی است (;