

① insert + selection این الگوریتم زمانی که بخش از آرایه مرتب است بهتر عمل می‌کند.

② خیر به ازای هر تقسیم بندر پیچیدگی  $n \log n$  است.

③ این آرایه تقریباً مرتب است

~~این آرایه تقریباً مرتب است~~

پس الگوریتم درجه مناسبی برای این کار است.

④ با پیچیدگی  $n \log n$  آن را مرتب می‌کنیم و سپس با پیچیدگی  $n$  تعداد کدها را آن را بدست می‌آوریم

$$O(n + n \log n) \leq O(n \log n)$$

⑤ این  $i = 0$   $a[i] \leq a[i+1]$  برود به

(س) اگر  $i$  صاف (۱- طول آرایه) برود به ۵

⑥  $i+1$  برود به ۴

⑦ آرایه را از نقطه  $i$  به دو قسمت تقسیم کرده و الگوریتم سوال ۳ را فیلد ۲ را اجرا می‌کنیم

$$O(n+k) \leq O(n)$$

ب) با  $n \log n$  آن را مرتب و با الگوریتم با  $n$  اعداد را پیدا می‌کنیم

$$O(n + n \log n) \leq O(n \log n)$$

سوال ۶- ۱ ۱۵۰

۲ مرتب ساز آرایه یک بعدی  $a[i]$  به روش دوم

۳ ۱۴۳

$$i \leftarrow i + a.length$$

int find second (array) {

if (length(array) < 2) return 0;

int max = 0;

int secmax = 0;

for (int i = 0; i < array.length; i++) {

if (~~number~~ <sup>a[i]</sup> > max;

~~secmax~~ secmax = max  
max = a[i]

else if: a[i] > secmax & a[i] != max;

secmax = a[i]

}  
}

~~if secmax ==~~

return secmax;

1 > max → max = 1, secmax = 0

0 > max → secmax = max, max = 0

1 < max → secmax = max, max = 1

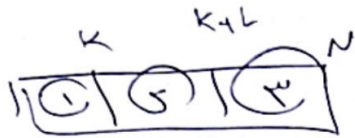
1 > secmax → —

~~1 > secmax~~

1 > secmax & 1 != max(1) → secmax = 1

1 > secmax → —

⇒ max = 1, secmax = 1



سوال ۹ 
$$\frac{N-K}{2} \leq L$$

برای مرتب شدن آرایه باید تابع  $is\_sorted$  پس از مرحله اول مرتب ساز شامل تمام  
 مکسیم ها را لازم باشد برابر با  $L$  داشته باشد و از داده ها بدویم می توان مطمئن شویم تابع  
 تعاد عناصر ما در یک عنصر از اعشار تابع  $is\_sorted$  دارد  $\frac{N-K}{2} \leq L$  بدین حال که در هر دو  
 داده معاد است.

سوال (۱۰) اگر  $\log n > K$  آرایه را مرتب کرده و  $K$  عنصر ابتدای آن را بر می داریم  
 اگر  $\log n < K$  ،  $K$  مرحله اول مرتب ساز عناصر را انجام می دهیم .