

-۱

	بدترین	میانگین	بهترین
Insertion Sort	$n^2$	$n^2$	$n$
Bubble Sort	$n^2$	$n^2$	$n$
Merge Sort	nlogn	nlogn	nlogn
Quick Sort	$n^2$	nlogn	nlogn
Heap Sort	nlogn	nlogn	nlogn

آرایه مرتب شده است پس Bubble sort و Insertion sort بهتر هستند.

-۲- حداکثر  $r + p - 1$  مقایسه لازم است.

-۳

- الف) نادرست  
 ب) درست  
 ج) نادرست  
 د) درست  
 ه) درست
- با توجه به سوال ۱ در بدترین حالت Merge sort بهتر عمل میکند.  
 در بدترین حالت Binary Insertion بهتر عمل میکند.  
 با توجه به سوال ۱ در بدترین حالت Heap sort بهتر عمل میکند.

-۴- نمیدونم والا

-۵- باید از ساختمان داده یک درخت متوازن استفاده کنیم؛ مثلاً باید از Red Black Tree استفاده شود.

```

Heapify (int i, Array[] arr)

    if (i >= arr.size())

        return;

    else

        Heapify (2 * i, arr);

        Heapify(2 * i + 1, arr);

        Swap (arr[i], min (arr[i], arr[2 * i], arr[2 * i + 1]));

```

```

FIND-MAXIMUM-SUBARRAY (A, low, high)
1  if high == low
2      return (low, high, A[low])           // base case: only one element
3  else mid =  $\lfloor (low + high) / 2 \rfloor$ 
4      (left-low, left-high, left-sum) =
          FIND-MAXIMUM-SUBARRAY (A, low, mid)
5      (right-low, right-high, right-sum) =
          FIND-MAXIMUM-SUBARRAY (A, mid + 1, high)
6      (cross-low, cross-high, cross-sum) =
          FIND-MAX-CROSSING-SUBARRAY (A, low, mid, high)
7      if left-sum  $\geq$  right-sum and left-sum  $\geq$  cross-sum
8          return (left-low, left-high, left-sum)
9      elseif right-sum  $\geq$  left-sum and right-sum  $\geq$  cross-sum
10         return (right-low, right-high, right-sum)
11     else return (cross-low, cross-high, cross-sum)

```

FIND-MAX-CROSSING-SUBARRAY(*A, low, mid, high*)

```
1  left-sum =  $-\infty$ 
2  sum = 0
3  for i = mid downto low
4      sum = sum + A[i]
5      if sum > left-sum
6          left-sum = sum
7          max-left = i
8  right-sum =  $-\infty$ 
9  sum = 0
10 for j = mid + 1 to high
11     sum = sum + A[j]
12     if sum > right-sum
13         right-sum = sum
14         max-right = j
15 return (max-left, max-right, left-sum + right-sum)
```

Rectangular Snip