1. (10 points)
   1. Given the data a[0:5] =[13, 2, 18, 5, 6, 9], please show the passes of insertion sort. Take as many rows as you need.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| pass | 13 | 2 | 18 | 5 | 6 | 9 |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |
| … |  |  |  |  |  |  |
| … |  |  |  |  |  |  |
| … |  |  |  |  |  |  |
| … |  |  |  |  |  |  |

* 1. What is the best-case complexity of insertion sort?
  2. What is the complexity of insertion sort if all elements are sorted except for one element?

1. (4 points) Given the data here a[1:10] =[13, 2, 18, 5, 7, 9, 4, 1, 3, 6]. Using the pseudocode in our class nodes (available on canvas module4), please write the result of maxHeapify(a, 2). Please note you may not use all rows, or please add more lines if needed.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 13 | 2 | 18 | 5 | 7 | 9 | 4 | 1 | 3 | 6 |
|  |  |  |  |  |  |  |  |  |  |

1. (6 points) Apply mergesort to a[1:8] = [3, 1, 4, 1, 5, 9, 2, 6].
   1. Determine the worst-case time complexity O() of mergesort for reverse-ordered input with size n.
   2. Illustrate the data flow status using the recursive example that we discussed in class.