CSE12 - Lecture 13

Wednesday, May 3, 2023 8:00 AM

PA4
PA1 Late / Resubmit > due tonight

PA5 released → due Tuesday

Quick Sort → in-place

Sorting Quickly

```
public class SortQuickly {

  public static void swap(String[] array, int i1, int i2) {
    String temp = array[i1];
    array[i1] = array[i2];
    array[i2] = temp;
  }

  public static int partition(String[] array, int low, int high) {
    int pivotStartIndex = high - 1;
    String pivot = array[pivotStartIndex];
    int smallerBefore = low, largerAfter = high - 2;

    while (smallerBefore <= largerAfter) {
      if (array[smallerBefore].compareTo(pivot) < 0) {
        smallerBefore += 1;
      }
      else {
        swap(array, smallerBefore, largerAfter);
        largerAfter -= 1;
      }
    }

    swap(array, smallerBefore, pivotStartIndex);
    return smallerBefore;
  }

  public static void qsort(String[] array, int low, int high) {
    if (high - low <= 1) { return; }
    int splitAt = partition(array, low, high);
    qsort(array, low, splitAt);
    qsort(array, splitAt + 1, high);
  }

  public static void sortD(String[] array) {
    qsort(array, 0, array.length);
  }

  public static void main(String[] args) {
    String[] str = {"f", "b", "a", "e", "d", "c" };
    int[] result = SortQuickly.sortD(str);
    System.out.println(Arrays.deepToString(result));
  }
}
```

Θ(1)

N 0 N

smaller | larger

Θ(N) N

ignored

["a", "b", "c", "d", "e", "f"]

Draw the picture of sortD()

What is the tight bound of sortD:

Best case: median value Θ(N * log2(N))

Worst case: sorted array Θ(N²)

Name: _____ PID: _____ Code: 2970

size = 6

low = 0
high = 6

0	1	2	3	4	5
f	b	a	e	d	c

pivotStartIndex = 5
pivot = "c"
sb = 0 1 2
la = 4 3 2 1

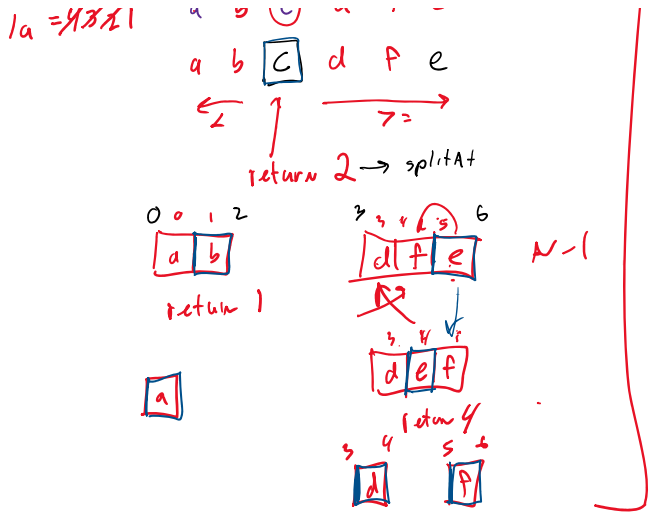
d	b	a	e	f	c
e	b	a	d	f	c
a	b	e	d	f	c
a	b	c	d	f	e

N

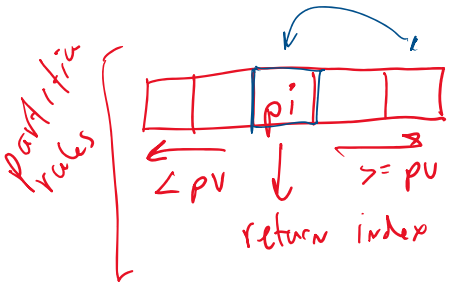
N = 6

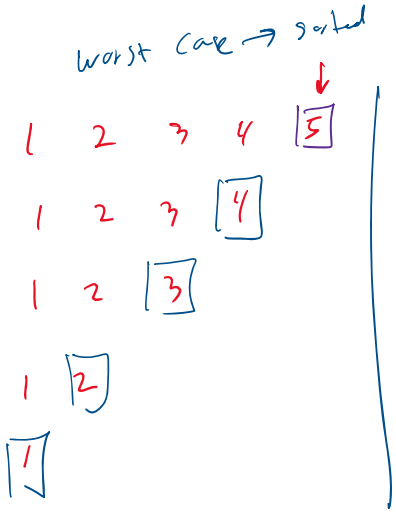
height = 3

↳ log2(N)



$\Theta(N * \log_2(N))$





$N * N \rightarrow \Theta(N^2)$