CSE12 - Lecture 13 - C00

Monday, October 24, 2022 11:00 AM

PA4 due tomorrow
PA1 Late/Resubmit → due tomorrow

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));
  }
}
```

$\Theta(1)$ 3

→ don't stop until sb is larger than la

$\Theta(N)$ N-2

$\Theta(1)$

$\Theta(1)$

0 6

6 0 6

1 2 3 4 5
sb la

Draw the picture of sortD()

What is the tight bound of sortD:

Best case: $\Theta(N * \log_2(n))$ → pivot is median value at all levels

Worst case: $\Theta(n^2)$ → sorted array

low = 0
high = 6

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

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

sb la

d b a e f c

e b a d f c

a b e d f c

la sb

a	b	c	d	f	e

← return index 2 →

a	b		d	f	e

a		d	e	f

d		f

a	b	c	d	e	f

1

2

3

N = 6

levels = 3

height = 3

$\log_2(N)$

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

worst case → sorted array

1	2	3	4	5

1	2	3	4

1	2	3

1	2

1

N = 5

levels = 5

$\Theta(N * N)$

$\Theta(n^2)$

valid partition

partition method

	pi	

< pv > pv

return pivot index

median value for the pivot

1	2	4	5	3

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

1	2	3	4	5