CSE12 - Lecture 12

Monday, April 29, 2024 10:00 AM

PA4 due Wednesday @ 8am
PA2 Late/Resubmit due Wednesday @ 8am

Lecture 12

Sorting Fast Merge Sort → divide & conquer algorithm

```
public class SortFast {

  public static String s(int[] arr) { return Arrays.toString(arr); }

  public static int[] combine(int[] part1, int[] part2) {
    int index1 = 0, index2 = 0;
    int[] combined = new int[part1.length + part2.length];
    while(index1 < part1.length && index2 < part2.length) {
      if(part1[index1] < part2[index2]) {
        combined[index1 + index2] = part1[index1];
        index1 += 1;
      }
      else {
        combined[index1 + index2] = part2[index2];
        index2 += 1;
      }
    }
    while(index1 < part1.length) {
      combined[index1 + index2] = part1[index1]; index1 += 1;
    }
    while(index2 < part2.length) {
      combined[index1 + index2] = part2[index2]; index2 += 1;
    }
    System.out.println(s(part1) + " " + s(part2) + " -> " + s(combined));
    return combined;
  }

  public static int[] sortC(int[] arr) {
    if(arr.length <= 1) { return arr; }
    else {
      int[] part1 = Arrays.copyOfRange(arr, 0, arr.length / 2);
      int[] part2 = Arrays.copyOfRange(arr, arr.length / 2, arr.length);
      System.out.println(s(arr) + " -> " + s(part1) + " " + s(part2));
      int[] sortedPart1 = sortC(part1);
      int[] sortedPart2 = sortC(part2);
      int[] sorted = combine(sortedPart1, sortedPart2);
      return sorted;
    }
  }

  public static void main(String[] args) {
    int[] result = SortFast.sortC(new int[] { 34, 93, 12, 49, 69, 25, 39 });
    System.out.println(SortFast.s(result));
  }
}
```

Draw the picture of sortC()

What is the tight bound of sortC:

Best case: Worst case:

Name: _____ PID: _____ Code: 6229

$2 + 2(m+n) + (m+n) + 1$

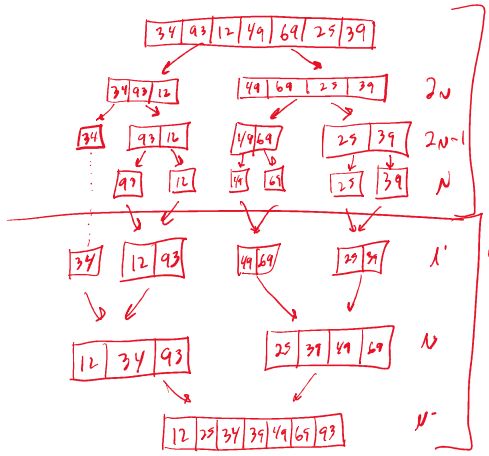
$3m + 3n + 3$

$\Theta(m+n) \rightarrow \Theta(N)$ combine/merge

$2N$

$2N-1$

copyOfRange → N → $2N$
↳ new array $\frac{N}{2}$ → N
↳ copy elements $\frac{N}{2}$ → N



splitting
$N = 7$
height = 3
$\lfloor \log_2(N) \rfloor$

combining/merging
height = 3
$\lfloor \log_2(N) \rfloor$

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