CSE12 - Lecture 12

Monday, May 1, 2023 8:00 AM

Sorting Fast

merge sort → divide & conquer

```
public class SortFast {

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

  public static int[] merge(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));
  }
}
```

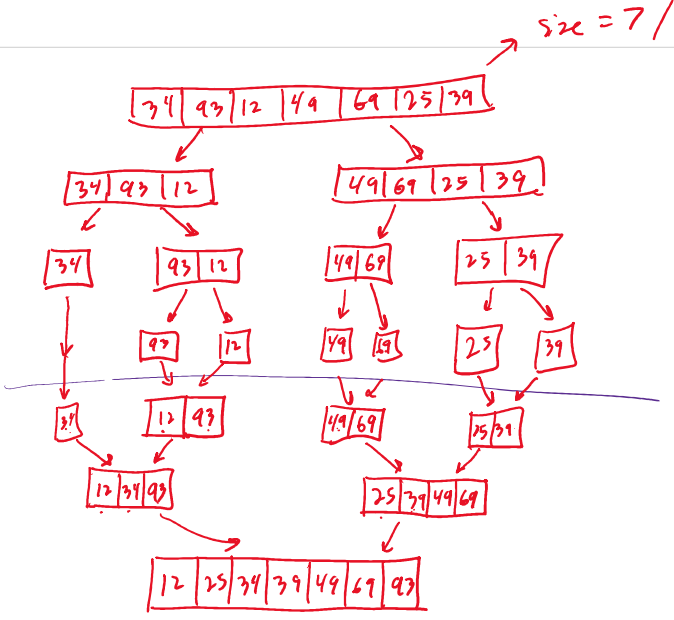
mergeSort

Draw the picture of sortC()

What is the tight bound of sortC:

Best case: $\Omega(N \cdot \log_2(N))$ Worst case: $O(N \cdot \log_2(N))$

Name: _____ PID: _____ Code: 5204



$2(N+m) + (N+m)2 + 4$

$4N + 4m + 4$

$O(N+m) \rightarrow \Theta(N)$

copy Of Range → 2N
↳ new array N
↳ copies the value N

Not in place → needs extra Memory

$N = 7$
$\log_2(7) = 3$

$N = 1000$
$\log_2(N) = 10$

size = 7/2 → 3

splitting

$N = 7$
height = 3
↳ $\log_2(N)$

$2N$
$2N-1$
$N-1$

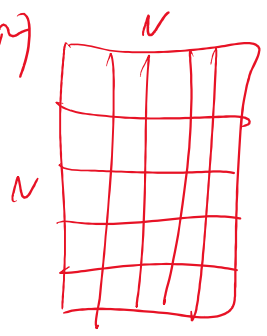
N
N
N

Combining/Merging
height = 3
↳ $\log_2(N)$

$2(N * \log_2(N))$

$N *$ height

$\Theta(N * \log_2 N)$



$O(N^2)$