Lecture 23

CSE 331

Oct 23, 2019

Mergesort algorithm

Input: $a_1, a_2, ..., a_n$ Output: Numbers in sorted order

MergeSort(a, n)
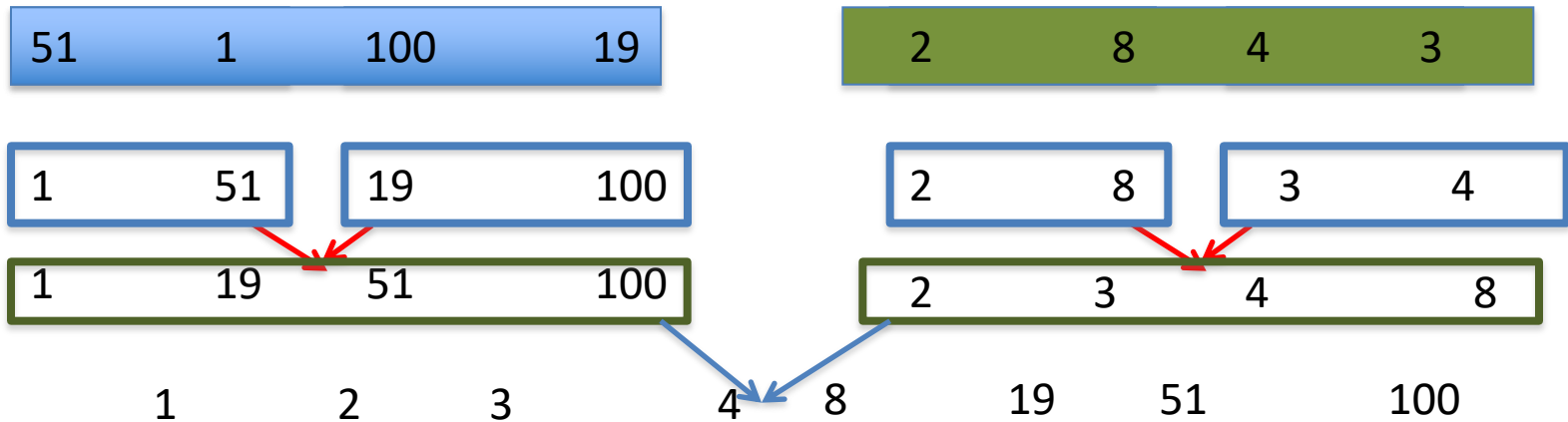
    If $n = 1$ **return** the order a_1

    If $n = 2$ **return** the order $\min(a_1,a_2); \max(a_1,a_2)$

    $a_L = a_1,..., a_{n/2}$

    $a_R = a_{n/2+1},..., a_n$

    return MERGE $\left(\text{MergeSort}(a_L, n/2), \text{MergeSort}(a_R, n/2) \right)$

An example run

51	1	100	19

2	8	4	3

1	51

19	100

2	8

3	4

1	19	51	100

2	3	4	8

1   2   3   4 8   19   51   100

MergeSort(a, n)

  If n = 1 **return** the order a_1
  If n = 2 **return** the order $\min(a_1, a_2)$; $\max(a_1, a_2)$

$a_L = a_1, \ldots, a_{n/2}$

$a_R = a_{n/2+1}, \ldots, a_n$

return MERGE (MergeSort(a_L, n/2), MergeSort(a_R, n/2))

Correctness

Input: $a_1, a_2, ..., a_n$ Output: Numbers in sorted order

MergeSort(a, n)

   If $n = 1$ **return** the order a_1
   If $n = 2$ **return** the order $\min(a_1, a_2); \max(a_1, a_2)$

   $a_L = a_1, ..., a_{n/2}$

   $a_R = a_{n/2+1}, ..., a_n$

   return MERGE (MergeSort($a_L, n/2$), MergeSort($a_R, n/2$))

By induction on n

Inductive step follows from correctness of MERGE

Rest of today's agenda

Analyze runtime of mergesort algorithm