

Lecture 27

CSE 331

Nov 4, 2016

HW 5 grading and the like

note stop following 4 views

UB Hacking and this weekend

As y'all know [UB Hacking](#) is this weekend and it should be great!

One thing to note with regard to 331 is that many of your wonderful TAs are part of UB Hacking organizing team so that means our response over the weekend would be slower than usual.

Also HW 5 grading will be delayed till early next week (though we should be back on track HW5 onwards).

[homework0](#)

[edit](#) | good note | 0

Updated 2 minutes ago by Adri Rudra

HW 7 posted

Homework 7

Due by **12:30pm, Friday, November 11, 2016.**

Make sure you follow all the [homework policies](#).

All submissions should be done via [Autolab](#).

Question 1 (Programming Assignment) [40 points]

`</>` Note

This assignment can be solved in either Java, Python or C++ (you should pick the language you are most comfortable with). Please make sure to look at the supporting documentation and files for the language of your choosing.

The Problem

In this problem, we will explore weighted graphs.

Solutions for HW 6

At the END of the lecture

Mergesort algorithm

Input: a_1, a_2, \dots, a_n

Output: Numbers in sorted order

```
MergeSort( a, n )
```

```
  If  $n = 1$  return the order  $a_1$ 
```

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

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

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

Correctness

Input: a_1, a_2, \dots, a_n

Output: Numbers in sorted order

MergeSort(a, n)

If $n = 1$ return the order a_1

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

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

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

Divide and Conquer

Divide up the problem into at least two sub-problems

Recursively solve the sub-problems

“Patch up” the solutions to the sub-problems for the final solution

Improvements on a smaller scale

Greedy algorithms: exponential \rightarrow poly time

(Typical) Divide and Conquer: $O(n^2)$ \rightarrow asymptotically smaller running time

Multiplying two numbers

Given two numbers a and b in binary

$$a = (a_{n-1}, \dots, a_0) \text{ and } b = (b_{n-1}, \dots, b_0)$$

Compute $c = a \times b$

Running time
of primary
school
algorithm?