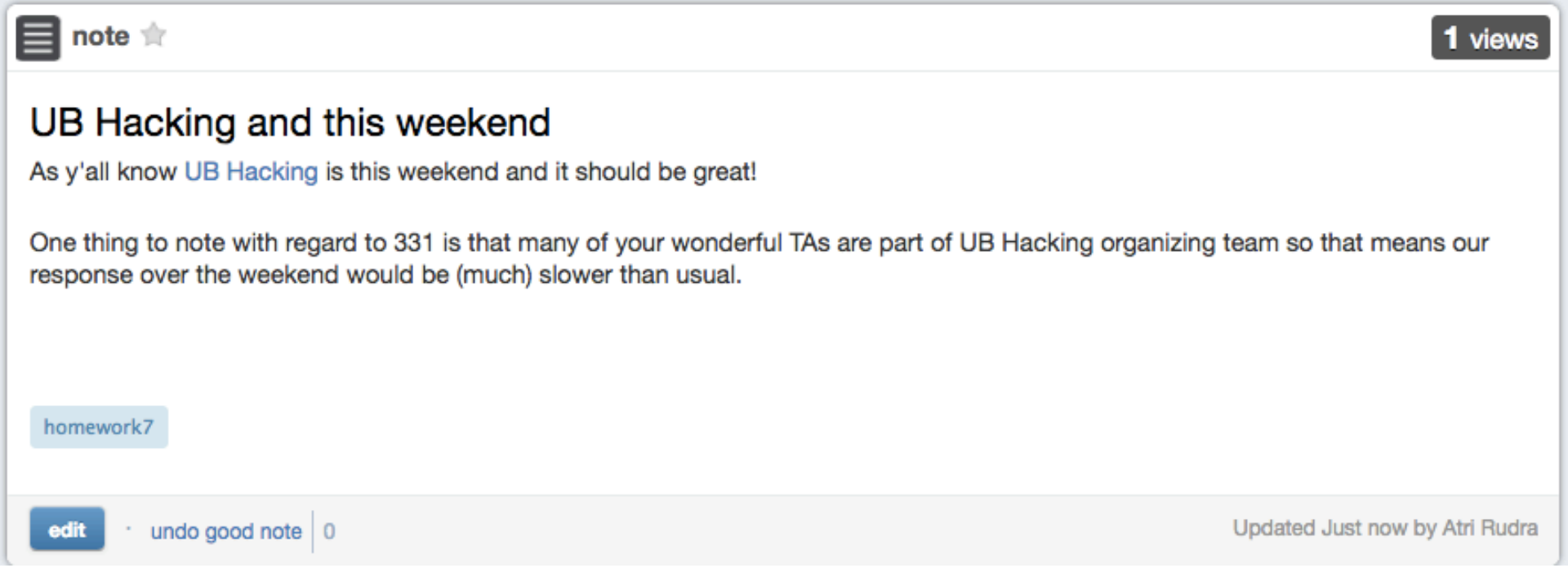
Lecture 27

CSE 331

Nov 3, 2017

UB Hacking


note
1 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 (much) slower than usual.
homework7
edit
undo good note
0
Updated Just now by Atri Rudra

You need to re-form groups

note

stop following 41 views

Video submission now open on Autolab

Sorry, forgot to do this earlier: you can now submit your video (note still PDF with the link in it) on Autolab.

YOU WILL NEED TO FORM YOUR GROUP ON AUTOLAB AGAIN BEFORE SUBMITTING.

See the mini project page for the details:

http://www-student.cse.buffalo.edu/~atri/cse331/fall17/mini-project/index.html

#pin

mini_project

edit · good note | 0

Updated 3 hours ago by Atri Rudra

HW 7 posted

Homework 7

Due by **11:00am, Friday, November 10, 2017**.

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.

We are given a starting node s and an ending node e, for some undirected graph G with n nodes. Further, each node u has its own weight, w_u `(0 <= w_u <= 50)`. The graph

Solutions for HW 6

At the END of the lecture

Mergesort algorithm

Input: $a_1, a_2, \ldots, a_n$

Output: Numbers in sorted order

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

    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) )
```

Correctness

Input: $a_1, a_2, \ldots, a_n$

Output: Numbers in sorted order

MergeSort(a, n)

If n = 1 **return** the order a_1

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

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

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

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 → poly time

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

Multiplying two numbers

Given two numbers a and b in binary

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

Compute c = a x b