Lecture 29

CSE 331

Nov 9, 2018

HW 8 out

Homework 8

Due by **11:59pm, Thursday, November 15, 2018**.

Make sure you follow all the homework policies.

All submissions should be done via Autolab.

Question 1 (Programming Assignment) [30 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 minimum spanning trees.

We are given a undirected, connected graph represented by its **adjacency matrix** representation. Our goal it to find a minimum spanning tree of that graph

HW 7 solutions

At the END of the lecture

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 $b = (b_{n-1},...,b_0)$

Compute $c = a \times b$

Elementary school algorithm is $O(n^2)$

The current algorithm scheme

Shift by O(n) bits

Adding O(n) bit numbe

Mult over n bits

$$a \cdot b = a^1b^1 \cdot 2^{2[n/2]} + (a^1b^0 + a^0b^1) \cdot 2^{[n/2]} + a^0b^0$$

Multiplication over n/2 bit inputs

$T(n) \leq 4T(n/2) + cn$

$T(1) \leq c$

T(n) is $O(n^2)$

The key identity

$$a^1b^0+a^0b^1 = (a^1+a^0)(b^1+b^0) - a^1b^1 - a^0b^0$$

The final algorithm

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

Mult (a, b)

If $n = 1$ return $a_0 b_0$

$a^1 = a_{n-1},...,a_{[n/2]}$ and $a^0 = a_{[n/2]-1},..., a_0$

Compute b^1 and b^0 from b

$x = a^1 + a^0$ and $y = b^1 + b^0$

Let p = Mult (x, y), D = Mult (a^1, b^1), E = Mult (a^0, b^0)

$F = p - D - E$

return $D \bullet 2^{2[n/2]} + F \bullet 2^{[n/2]} + E$

$T(1) \leq c$

$T(n) \leq 3T(n/2) + cn$

$O(n^{\log_2 3}) = O(n^{1.59})$ run time

All green operations are $O(n)$ time

$$a \bullet b = a^1b^1 \bullet 2^{2[n/2]} + ((a^1+a^0)(b^1+b^0) - a^1b^1 - a^0b^0) \bullet 2^{[n/2]} + a^0b^0$$