Analysis of Algorithms - CS 323
Lecture #14 – May 18, 2016

Notes by: Anna Ip

Announcements

The final exam will start at 6:30 p.m. on Wednesday, May 25th.

You can submit today's lecture notes for extra credit (2 points or 3 if it's really good) by 11 a.m. on Sunday, May 22nd.

Lecture Notes

Countable → there is a mapping of every element in a set to the natural numbers ($\mathbb{N}$)

Uncountable → you will miss some numbers in the set ($\mathbb{R}$), Cantor's Diagonalization Argument

Inclusive (up to and including) → [0, 1) ← Exclusive (up to but not including)

0 . $\mathbf{b_{1,1}}$ $b_{1,2}$ $b_{1,3}$ $b_{1,4}$ . . .
0 . $b_{2,1}$ $\mathbf{b_{2,2}}$ $b_{2,3}$ $b_{2,4}$ . . .
0 . $b_{3,1}$ $b_{3,2}$ $\mathbf{b_{3,3}}$ $b_{3,4}$ . . .

Look at the diagonal values and take the complement

$\bar{b}_{i,i} = 1 - \bar{b}_{i,i}$

0 → 1
1 → 0

$r^* = 0 . \bar{b}_{1,1}\ \bar{b}_{2,2}\ \bar{b}_{3,3}\ \bar{b}_{4,4} \ldots$

f(x) → y

$x \in \mathbb{N}$

$y \in \mathbb{N}$

x	$f_0(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$	...	$f^*(x)$
0	$f_0(0)$	$f_1(0)$	$f_2(0)$	$f_3(0)$		$f_0(0) + 1$
1	$f_0(1)$	$f_1(1)$	$f_2(1)$	$f_3(1)$		$f_1(1) + 1$
2	$f_0(2)$	$f_1(2)$	$f_2(2)$	$f_3(2)$		$f_2(2) + 1$
3	$f_0(3)$	$f_1(3)$	$f_2(3)$	$f_3(3)$		$f_3(3) + 1$
4	$f_0(4)$	$f_1(4)$	$f_2(4)$	$f_3(4)$		...
...	...	...	...	...		$f_i(i) + 1$

There are countably infinite programs
There are uncountably infinite simple integer functions

The Halting Problem

```
program(x)
    if halt(x, x)
        then loop forever
    else
        halt
```

The Median Problem

NP Complete Problems

- IS (Independent Set) Problem
    - Subgraph S such that no edges in V_s are adjacent
- VC (Vertex Cover) Problem
    - Subgraph S such that each $v_i \in V_s$ is adjacent to another vertex $v_i \in V_s$

IS and VC are complements of each other

Final Exam Review

Time complexities
Names of main algorithms

1 question from the first exam and 1 question from the second exam (more general)

For example:

1) Sorting algorithms
2) Graph algorithms
3) Pseudocode for something ← e.g., Halting problem

The rest are from the last 1.5 lectures

Sequences (know the most basic ones, like these)

$1 + 2 + 3 + \ldots + n = n(n+1)/2$

$r + r^2 + r^3 + \ldots + r^n = (r^{n+1} - 1)/(r - 1)$

$f(n) = f(n) + f(n - 2)$ ← Characteristic equation (be able to identify what this is)
$f(n) - f(n - 1) - f(n - 2) = 0$
$x^2 - x - 1 = 0$

$n = 2^k$
$T(n) = T(n/2) + f(n)$ ← Domain transformation (change the input)
$S(k) = S(k - 1)$

$T(n) = 2T(n/2) + \ldots$ ← Range transformation (change the output)

Master theorem ← What theorem helps algorithms estimate time complexity

Dynamic programming technique
- Floyd
- Dijkstra
- Prim
- Bellman Ford ← Handles negative weight complexity, O(V * E)

~~Weighted triangulation~~ ← Not on the final

Dynamic problem solution

Knapsack problem ← Classic NP complete problem, $O(n * k)$ where k is the upper bound on weight capacity of knapsack

Classic knapsack problem → You want to maximize the value

0-1 knapsack problem

0-1 knapsack problem with integer capacity → $O(n * k)$ but k will be very large to n, more like exponential

Fractional knapsack problem → $O(n)$, sort by: (value/weight)

Basic matrix multiplication problem

Divide and conquer approach, divide into 4 quadrants

Now you have 8 subproblems, solve those

$T(n) = 8T(n/2) + O(n^2)$

$T(n) = O(n^3)$

Strassen

$T(n) = 7T(n/2) + O(n^2)$

$T(n) = O(n^{\log_2 7})$