Lecture 19

11/23

Divide & Conquor RR

Sps a recursive algo divides a prob of size n into subprobs each of size n/b. Also sps $g(n)$ extra ops needed in conquor step. If $f(n)$ reps the # ops required to solve prob of size n, then

$$f(n) = a f(n/b) + g(n)$$ is called a Div & Conquor RR

Ex Binary Search

```
def binary-search(arr, l, r, x):
    if r ≥ l:
        mid = (r+l)//2
        if arr[mid] == x:
            return mid
        elif arr[mid] > x:
            return binary-search(arr, l, mid, x)
        elif arr[mid] < x:
            return binary-search(arr, mid, r, x)

    else:
        return -1
```

What is the complexity of binary search?

What is the complexity of binary search?

$\Rightarrow$ What is the RR?

$$T_n = T_{n/2} + 1 \quad \text{with } T_1 = 0$$

Since $T_{n/2} = T_{n/4} + 1 \Rightarrow T_n = (T_{n/4} + 1) + 1$

$$\Rightarrow T_n = (T_{n/2^3} + 1) + 2$$

$$= T_{n/2^3} + 3$$

What happens at the n^{th} step?

$T_n = T_{n/2^r} + r \Rightarrow$ Now $T_1 = 0$ so $n/2^r = 1$

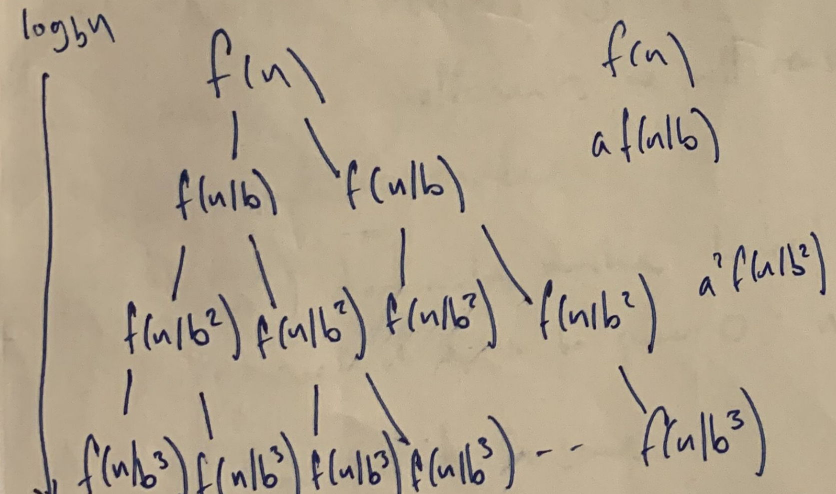
$$\Rightarrow n = 2^r$$

$$\Rightarrow r = \log n$$

$$\Rightarrow T_n = T_1 + \log n$$

$$= \log n$$

$\Rightarrow$ Binary search is $O(\log n)$



We can draw a tree generated by the RR. Tree has depth $\log_b n$ and branching factor a. There are a^i nodes at level i, each labeled $f(n/b^i)$. The value of $T(n)$ is the sum of ~~all~~ the labels of all the nodes in the tree.

A unifying method for solving div & conquer RR

<u>Thm</u> Let f be an increasing function that satisfies RR

$f(n) = a f(n/b) + cn^d$, whenever $n = b^k$ where $k \in \mathbb{Z}^+$

$a \geq 1, b \in \mathbb{Z}, b > 1, c, d \in \mathbb{R}$ with $c \in \mathbb{R}^+, d > 0$

$$f(n) = \begin{cases} O(n^d) & \text{if } a < b^d \quad \text{case I} \\ O(n^d \log n) & \text{if } a = b^d \quad \text{case II} \\ O(n^{\log_b a}) & \text{if } a > b^d \quad \text{case III} \end{cases}$$

$T(n) = a \cdot T(n/b) + O(n^d)$ general formula for div & Conquer algo

or $f(n) = a f(n/b) + g(n)$

By looking at coefficients we will reason about which part of RR will dominate the runtime.

I) $O(n^d)$ dominates runtime. Level 0 will dominate runtime. coef of a is to power 0. n^d got cancelled out and is just const term. in Binary search, splitting did not take long.

II) All levels will be the same.

III) Lowest level h will dominate runtime.

Total runtime $T(n) = 1 \cdot n^d + a\left(\frac{n}{b}\right)^d + a^2\left(\frac{n}{b^2}\right)^d + \dots + a^h\left(\frac{n}{b^h}\right)^d$

Total amount of work

$\Rightarrow$ pull out n^d $\Rightarrow n^d\left(1 + a\left(\frac{1}{b}\right)^d + a^2\left(\frac{1}{b^2}\right)^d + \dots + a^h\left(\frac{1}{b^h}\right)^d\right)$

$= n^d\left[1 + \frac{a}{b^d} + \left(\frac{a}{b^d}\right)^2 + \dots + \left(\frac{a}{b^d}\right)^h\right]$ $\quad \frac{a}{b^d} \gtrless 1$

Geometric series.

Note $\frac{1}{1-r} = 1 + r + r^2 + \cdots + r^h$

Case I if $a \cancel{<} b^d$ then $r = \frac{a}{b^d} < 1$

$\Rightarrow T(n) = O(n^d - 1) = O(n^d)$

Case II if $a = b^d$ then $r = 1 \Rightarrow$ All terms are 1

$\Rightarrow h+1$ terms

$\Rightarrow T(n) = O(n^d(h+1)) = O(n^d \cdot h)$

we know $n = b^h$, so $h = \log_{base} n = O(\log n)$

$\Rightarrow O(n^d \cdot \log n)$

Case III if $a > b^d$ then $T(n) = O\left(n^d \left(\frac{a}{b^d}\right)^h\right) = O(a^h)$

$= O(a^{\log_b n}) = O(n^{\log_b a})$ via log prop

Lemma $a^{\log_b n} = n^{\log_n a \log_b n} =$

Note $\log_n a = \frac{\log_b a}{\log_b n} \Rightarrow n^{\log_n a \log_b n} = n^{\log_b a}$.

Def the generating function for the seq $a_0, a_1, \ldots, a_k$ of real #s is the infinite series

$$G(x) = a_0 + a_1x + a_2x^2 + \ldots a_kx^k + \ldots = \sum_{k=0}^{\infty} a_k x^k$$

Generating functions used to represent sequences efficiently by coding terms of a seq as coeffs of powers of variable x in a formal power series.

Can be used to solve counting problems, RR's by translating RR for terms of a seq into an equation.

The generating function for the sequence

Ex $\{a_k\}$, $a_k = 3$, $a_k = k+1$, $a_k = 2^k$

$$\sum_{k=0}^{\infty} 3x^k \qquad \sum_{k=0}^{\infty} (k+1)x^k \qquad \sum_{k=0}^{\infty} 2^k x^k$$

Ex Let $a_k = \binom{M}{k}$ for $k = 0, 1, \ldots, M$. What is gen fn for $a_0, \ldots a_M$

$$G(x) = C(M,0) + C(M,1)x + C(M,2)x^2 + \ldots + C(M,M)x^M$$

$$\binom{M}{0} + \binom{M}{1}x + \binom{M}{2}x^2 + \ldots + \binom{M}{M}x^M$$

$$= (1+x)^M \text{ by Binomial Thm.}$$

Thm Let $f(x) = \sum_{k=0}^{\infty} a_k x^k$, $g(x) = \sum_{k=0}^{\infty} b_k x^k$

Then $f(x) + g(x) = \sum_{k=0}^{\infty} (a_k + b_k) x^k$

$$f(x) g(x) = \sum_{k=0}^{\infty} \left(\sum_{j=0}^{k} a_j b_{k-j} \right) x^k$$

only valid for
power series that converge in an interval.

Using Generating functions to solve RR

Ex $a_k = 3a_{k-1}$ for $k = 1, 2, 3, \ldots$, $a_0 = 2$

Let $G(x)$ be the generating function for the ~~seq~~ $\{a_n\}$

$$G(x) = \sum_{k=0}^{\infty} a_k x^k$$

Note $x G(x) = \sum_{k=0}^{\infty} a_k x^{k+1} = \sum_{k=1}^{\infty} a_{k-1} x^k$

Using RR

$$G(x) - 3x G(x) = \sum_{k=0}^{\infty} a_k x^k - 3 \sum_{k=1}^{\infty} a_{k-1} x^k$$

$$= a_0 + \sum_{k=1}^{\infty} \underbrace{(a_k - 3a_{k-1})}_{0 \text{ since } a_k = 3a_{k-1}} x^k$$

$$= 2$$

So $G(x) - 3x G(x) = 2$

$G(x) = 2/(1-3x)$

Using identity

$1/(1-ax) = \sum_{k=0}^{\infty} a^k x^k$

$G(x) = 2 \sum_{k=0}^{\infty} 3^k x^k$

$= \sum_{k=0}^{\infty} 2 \cdot 3^k x^k$

$\Rightarrow a_k = 2 \cdot 3^k$

Ex $a_n = 8a_{n-1} + 10^{n-1}$

$a_1 = 9$

Use Generating fn to find explicit formula for a_n

$\Rightarrow a_1 = 8a_0 + 10^0 = 8+1 = 9$

$\Rightarrow a_n x^n = 8a_{n-1}x^n + 10^{n-1}x^n$

Let $G(x) = \sum_{n=0}^{\infty} a_n x^n$ be the generating funct of the seq $a_0, a_1, \ldots a_n$

$$G(x) - 1 = \sum_{n=1}^{\infty} a_n x^n = \sum_{n=1}^{\infty}\left(8a_{n-1}x^n + 10^{n-1}x^n\right)$$

$$= 8\sum_{n=1}^{\infty} a_{n-1}x^n + \sum_{n=1}^{\infty} 10^{n-1}x^n$$

$$= 8x\sum_{n=1}^{\infty} a_{n-1}x^{n-1} + x\sum_{n=1}^{\infty} 10^{n-1}x^{n-1}$$

$$= 8x\sum_{n=0}^{\infty} a_n x^n + x\sum_{n=0}^{\infty} 10^n x^n$$

$$= 8x\,G(x) + x/(1-10x)$$

since $f(x) = \frac{1}{1-ax}$ is gen func for $1, a, a^2, a^3$

$1/_{1-ax} = 1 + ax + a^2x^2 + \ldots$

$$G(x) - 1 = 8x\,G(x) + \frac{x}{1-10x}$$

$$G(x) = \frac{1-9x}{(1-8x)(1-10x)}$$

$$G(x) = \frac{1-9x}{(1-8x)(1-10x)}$$

Trick: expand RHS into partial fractions (as done in integration of rational functions)

$$G(x) = \frac{1}{2}\left[\frac{1}{1-8x} + \frac{1}{1-10x}\right] \quad \text{Since } \frac{(1-10x)+(1-8x)}{2(1-8x)(1-10x)}$$

Using $f(x) = \dfrac{1}{1-ax}$ twice $\quad = \dfrac{2-18x}{2(1-8x)(1-10x)} = \dfrac{1-9x}{}$

with $a=8$, $a=10$

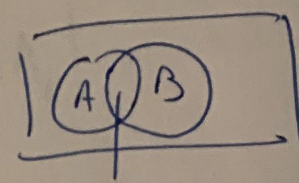
$$\Rightarrow G(x) = \frac{1}{2}\left(\sum_{n=0}^{\infty} 8^n x^n + \sum_{n=0}^{\infty} 10^n x^n\right)$$

$$= \sum_{n=0}^{\infty} \frac{1}{2}(8^n + 10^n)x^n \Rightarrow a_n = \frac{1}{2}(8^n + 10^n)$$

Principle of Inclusion-Exclusion

How many elements in intersection of two sets?

$$|A \cup B| = |A| + |B| - |A \cap B|$$



$A \cap B$

Ex. 1807 freshmen,

of which 453 taking CS, 567 taking math, 299 taking both.

How many not taking either math or CS?

A be fresh taking CS

B " " " math

$|A| = 453$, $|B| = 567$, $|A \cap B| = 299$

$$|A \cup B| = |A| + |B| - |A \cap B|$$
$$= 453 + 567 - 299 = 721$$

So $1807 - 721 = 1086$ not taking CS or Math.

$\Rightarrow$ Can we generalize to a union of a finite # of ~~sets~~ n sets?

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

Principle of Inclusion-Exclusion

Let $A_1, A_2, \ldots, A_n$ be finite sets.

$$|A_1 \cup A_2 \cup \cdots \cup A_n| = \sum_{1 \le i \le n} |A_i| - \sum_{1 \le i < j \le n} |A_i \cap A_j|$$

$$+ \sum_{1 \le i < j < k \le n} |A_i \cap A_j \cap A_k| - \cdots + (-1)^{n+1} \times |A_1 \cap A_2 \cap \cdots \cap A_n|$$

Probability

<u>Def</u> An exp is a proc that yields <u>one</u> of a given set of possible outcomes. The <u>sample space</u> of the exp is the set of possible outcomes. An <u>event</u> is a subset of the SS.

If S is a finite nonempty SS of <u>equally likely</u> outcomes and E is an event (subset of S) then the prob of E

$$p(E) = \frac{|E|}{|S|}$$

Accordingly, prob of an event is between 0 and 1

$$0 \leq |E| \leq |S| \quad b/c \quad E \subseteq S$$

$$\Rightarrow 0 \leq p(E) = \frac{|E|}{|S|} \leq 1$$

<u>Ex</u>. prob of sum of the numbers on two dice rolled is 7?

$(1,6), (2,5), (3,4), (4,3), (5,2), (6,1) = \frac{6}{36} = \frac{1}{6}$

<u>Ex</u> Lottery asks to pick set of 6 #s out of first n positive integers, where n between 30 and 60. What is prob picking 6 correct out of 40?

$$C(40,6) = \frac{40!}{34!\,6!} = 3,838,380 \text{ is sample space}$$

$1/3,838,380$

Thm

Let E be event in SS S. The prob of $\bar{E} = S - E$ (the complement of E)

is $p(\bar{E}) = 1 - p(E)$

Pf. Note $|\bar{E}| = |S| - |E| \Rightarrow p(\bar{E}) = \frac{|S| - |E|}{|S|} = 1 - \frac{|E|}{|S|} = 1 - p(E)$

Ex. A seq of 10 bits randomly generated.
prob at least 1 is 0?

$1 -$ prob none are $0 = 1 - \frac{1}{2^{10}} = 1 - \frac{1}{1024} = \frac{1023}{1024}$

Thm Let E_1, E_2 be events in SS S.

$$P(E_1 \cup E_2) = P(E_1) + P(E_2) - P(E_1 \cap E_2)$$

Axioms

i) $0 \le p(s) \le 1 \quad \forall s \in S$

ii) $\sum_{s \in S} p(s) = 1$

when n possible outcomes

$0 \le p(x_i) \le 1 \quad i = 1, \ldots, n$

$\sum_{i=1}^{n} p(x_i) = 1$

The function p from the set of all possible outcomes of SS S is called a prob dist.