Divide & Conquer RR

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

\[ f(n) = af(n/b) + g(n) \]

is called a Divide & Conquer RR

Ex: Binary Search

```python
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)
        else:
            return binary_search(arr, mid + 1, r, x)

    return -1
```

What is the complexity of binary search?
What is the complexity of binary search?

What is the RB?

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

Since \[ T_{n/2} = T_n/4 + 1 \] \[ \Rightarrow T_n = (T_n/4 + 1) + 1 \]
\[ \Rightarrow T_n = (T_n/2^3 + 1) + 1 \]
\[ = T_n/2^3 + 3 \]

What happens at the \( n \)th step?

\[ T_n = T_{n/2} + r \quad \Rightarrow \text{Now} \quad T_1 = 0 \quad \text{so} \quad n/2^r = 1 \]

\[ \log_{n/2} n = f(n) \quad \Rightarrow r = \log n \]
\[ \Rightarrow n = 2^r \]
\[ \Rightarrow \text{Binary search is} \quad O(\log n) \]

We can draw a tree generated by the RB. Tree has depth \( \log_2 n \) and branching factor \( a \). There are \( a^i \) nodes at level \( i \), each labeled \( f(n/2^i) \). The value of \( T(n) \) is the sum of the labels of all the nodes in the tree.
A unifying method for solving divide-and-conquer RR

Thm: Let $f$ be an increasing function that satisfies RR

$$f(n) = a f(n/b) + c d^n$$

where $n = b^k$ where $k \in \mathbb{Z}^+$

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

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

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + O(n^d)$$

give divide and 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. Here $O$ will dominate runtime.

cof of $a$ is to power $O$. $a^d$ got cancelled out as we just cast it in binary search.

splitting did not take long.

II) All levels will be the same.

III) Honest level $h$ will dominate runtime.

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

Total amount of work

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

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

Geometric series.
Note: \[
\frac{1}{1-r} = 1 + r + r^2 + \ldots + r^n
\]

**Case I** if \( a \neq b^d \) then \( r = \frac{a}{b^d} < 1 \)

\[ T(n) = O(n^d - 1) = O(n^d) \]

**Case II** if \( a = b^d \) then \( r = 1 \) \( \rightarrow \) All terms are 1

\[ T(n) = O(n^d(1 + 1)) = O(n^d \cdot 1) \]

we know \( n = b^h \), so \( h = \log_b n \Rightarrow \log b^h = O(\log n) \)

\[ T(n) = O(n^d \cdot \log n) \]

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

\[ = O(a \log_b n) = O(n \log_2 a) \text{ via log prop} \]

\[ \text{Therefore} \quad a \log_b n = n \log_2 a \log_2 n = \]

Note: \[ \log_n a = \frac{\log_2 a}{\log_2 n} \Rightarrow n \log_2 a = n \log_2 n \log_2 n \]
Define the generating function as the sequence \( a_0, a_1, \ldots, a_n \) of real numbers as the infinite series
\[
G(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n = \sum_{k=0}^{\infty} a_kx^k
\]

Generating functions are used to represent sequences efficiently by coding terms of a sequence as coefficients of powers of variable \( x \) in a power series.

Can be used to solve combinatorial problems, e.g., by finding the power series of a sequence of numbers.

Example: \( S(a_n) \), \( a_0 = 3 \), \( a_n = kn \), \( a_n = 2^k \)

\[
\sum_{k=0}^{\infty} 3x^k = \sum_{k=0}^{\infty} (kn)x^k = \sum_{k=0}^{\infty} 2^k x^k
\]

Example: but \( a_n = \binom{m}{k} \) as \( n = 0, 1, \ldots, m \). What is \( \sum_{n=0}^{m} a_n \) as a generating function?

\[
G(x) = C(M,0)x^0 + C(M,1)x^1 + C(M,2)x^2 + \cdots + C(M,M)x^M
\]

\[
= (1+x)^M
\]

by Binomial Theorem.
Thm \( f(x) = \sum_{k=0}^{\infty} a_k x^k \), \( g(x) = \sum_{k=0}^{\infty} b_k x^k \)

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

\( f(x) g(x) = \sum_{k=0}^{\infty} \left( \sum_{i=0}^{k} a_i b_{k-i} \right) x^k \)

only valid for power series that converge uniformly.

Using Generating Functions to solve \( BR \)

\[ a_k = 3a_{k-1} \text{ for } k \geq 1, \ a_0 = 2 \]

Let \( G(x) \) be the generating function for \( a_k \).

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

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

Using \( BR \)

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

\[ = a_0 + \sum_{k=1}^{\infty} (a_k - 3a_{k-1}) x^k \]

\[ = 2 \quad \text{or} \quad a_0 = 3a_{k-1} \]
\[ a_n = 8a_{n-1} + 10^{n-1} \quad a_1 = 9 \]

Use generating fun 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 function of the seq \( a_0, a_1, \ldots \)

\[ G(x) - 1 = \sum_{n=1}^{\infty} a_n x^n = \sum_{n=1}^{\infty} (8a_{n-1} x^n + 10^{n-1} x^n) \]

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

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

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

\[ = 8x G(x) + \frac{x}{1 - 10x} \]

Since \( f(x) = \frac{1}{1 - ax} \) is generating for \( 1, a, a^2, a^3, \ldots \)

\[ G(x) - 1 = 8x G(x) + \frac{x}{1 - 10x} \]

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

**Trick:** expand R.H.S into partial fractions (as done in earlier problems).

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

Using \( f(x) = \frac{1}{1 - ax} \) twice

\[
G(x) = \frac{2 - 18x}{2 (1 - 8x)(1 - 10x)} = \frac{1 - 9x}{1 - 8x}(1 - 10x)
\]

with \( a = 8, \ b = 10 \)

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

\[
= \sum_{n=0}^{\infty} \frac{1}{2} \left( 8^n + 10^n \right) x^n \Rightarrow a_n = \frac{1}{2} \left( 8^n + 10^n \right)
\]
Principle of Inclusion-Exclusion

How many elements in intersection of two sets?

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

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 +/- CS

B be +/- 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.

Can we generalize to union of a finite number of 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.

\[
\left| A_1 \cup A_2 \cup \cdots \cup A_n \right| = \sum_{i=1}^{n} \left| A_i \right| - \sum_{1 \leq i < j \leq n} \left| A_i \cap A_j \right| + \sum_{1 \leq i < j < k \leq n} \left| A_i \cap A_j \cap A_k \right| - \cdots + (-1)^{n+1} \left| A_1 \cap A_2 \cap \cdots \cap A_n \right|
\]
Probability

Def An exp is a proc that yields one of a given set of possible outcomes. The sample space of the exp is the set of possible outcomes. An event is a subset of the ss.

If S is a finite nonempty ss of equally likely 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

\[ 0 \leq |E| \leq |S| \quad \text{b/c } E \subseteq S \]

\[ \Rightarrow 0 \leq P(E) = \frac{|E|}{|S|} \leq 1 \]

Ex. Prob of sum of two numbers on two dice rolled is 7?

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

Ex. Heaven asks to pick set of 6 #s out of list of possible integers, where n between 30 and 60. What is prob picks 6 correct out of 40?

\[ C(40,6) = \frac{40!}{34!6!} = 3,838,380 \quad \text{is some xpr} \]

\[ 1/3,838,380 \]
Let $E$ be an event in $SS$. The prob of $E=S-E$ (complement of $E$) is 

$$p(E) = 1 - p(E)$$

\[\text{Pr. Note } |E| = |S| - |E| \Rightarrow p(E) = \frac{|S| - |E|}{|S|} = 1 - \frac{|E|}{|S|} = 1 - p(E)\]

Ex. A seq of 10 bits randomly generated.

Prob of 1 ends 1 is $0$?

$$1 - \text{prob none-O} = 1 - \frac{1}{2^{10}} = 1 - \frac{1}{1024} = \frac{1023}{1024}$$

Then let $E_1, E_2$ be events in $SS$.

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

**Axioms**

i) $0 \leq p(S) \leq 1 \ \forall S \in S$

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

The func $p$ from the set of all possible events of $SS$ is called a prob dist.