Algorithms for BigData [CSL7630]

Kadapala Rakesh Reddy - M22AI608

minor-2.

IIT-J. M.Tech-DCS. sem-III.

4) Ans) (a) Monte carlo algorithms are preffered over Las vegas Algorithm even when soln may be incorrect due to following Reasons:

Efficency & scalability: Monte carlo algo. highly efficent & scalable. They can handle complex problems with large soln spaces & high dimentional scenario where deterministic approaches become impractical or computationally expensive.

It explores a vast no. of probabilities & provide approximate soln within a reasonable time frame.

Approximate solutions: while solns may not be exact they often offer a close estimation with a known level of confidence.

Probabilistic Insights: Monte carlo Algo. excel provides probabilistic insights. They can estimate & evaluate statistical measures & simulate random events. This makes them valuable in various fields including risk analytics, scientific simulations.

Trade of b/w correctness & efficency: monte carlo Algo strikes a Trade off b/w correctness & efficency while Las vegas Algo. guarantee correct soln. They might be less efficient due to their deterministic nature.

Real world Applicability: In May Real world scenarios an exact soln may might not necessary due to uncertainties & limitations.
So: Montecarlo Also Provides practical approach.

finally The Relevance of monte Carlo Also over Las vegas Algo. despite The possibility of Incorrect soln is driven by their Efficiency, scalability.

(4) (b) Binomial probability formula

$$P(X \geq k) = 1 - P(X < k)$$

$P(X < k)$ represents cumulative probability of getting fewer than k success

$k = 0$ (no success) we have to calculate probability for hitting bull eye. atleast ones.

The probability of hitting bullseyes in a single shot is $P = 0.6$
no. of Trials $n = 4$

$P(X < 1)$ (CDF) Binomial cumulative Distribution function

$$P(X < 1) = CDF(k-1, n, P)$$

$CDF(k-1, n, P)$ = cumulative Probability upto k-1 success

$$P(X < 1) = CDF(0, 4, 0.6)$$

$$P(X < 1) = 0.1296$$

$P(x \geq 1) = 1 - P(x < 1) = 1 - 0.1296$

$= 0.8704$

$= 88\%$

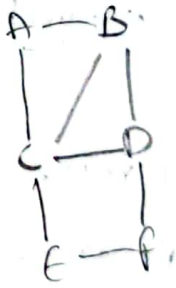
So Probability of succeeding in hitting bull's eye atleast once in four shot approximately 88%.

(2) The Randomized min-cut algorithm is based on concept of contraction.

It Repeatedly selects Random edges & contracts them until there are only 2 vertices left which Represents min cut of graph

(1) Vertices A, B, C, D, E, F

Edges (A,B), (A,C) (B,C), (B,D) (C,D) (C,E) (D,F) (E,F)



- Randomly selecting an edge (AC)
- Contract edge (A,C) merging vertices A & C into one vertex AC

Then graph becomes

Vertices AC, B, D, E, F

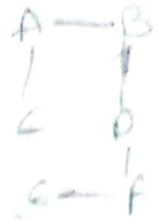
Edges : (AC,B) (B,D) (D,F) (E,F)

- again selecting random edge (AC, B)

contracting edge (AC, B) by merging vertices AC & B in 1 vertex

∴ AcB vertices AcB, D, E, F

edges (AcB, D) (D, F) (E, F)



- Repeating steps until there are only 2 vertices left
- Randomly select (D, F) edge

Contract edge (D, F) by merging vertices D, F in 1 vertex

vertices : AcB, DF, E

edges (AcB, DF) (AcB, E)

∴ The resulting graph has only 2 vertices AcB & E.

The cut b/w 2 vertices represents the graph min cut in original graph.

The resulting min. cut b/w vertices AcB & E. The benefit of using Randomized Algo. like Randomized min-cut Algo is that it has high probability of finding correct min. cut.

(1) case(i) The goat is choosen & another goat is revealed

Then chance of winning = 1.

case(ii) If goat is choosen & another goat is revealed switching the door & chance of winning = 1.

case(iii) If a gold bar is choosen, a goat is revealed switching & getting gold is $\frac{1}{2}$

case(iv) If a gold is choosen a goat is revealed. switching & getting gold is $\frac{1}{2}$

From above all cases, chance of winning is

$$1 + 1 + \frac{1}{2} + \frac{1}{2} = 3$$

But only one out of the 4 cases happens.

∴ chances of winning probability is $\frac{3}{4}$

3

3

To sort a string of n characters O(n) Time using a randomized Algorithm, we can utilize a modified Quick sort algorithm.

The key Idea is to choose a random pivot element & partition the string around that pivot

Base Case: if string has zero or one character it is already sorted so return string.

- Randomly select pivot character from string.
- Partitioning string into 3 parts. Small, equal & larger characters

Recursive sort the smaller & larger partitions using Algorithm

Concatenate the sorted smaller Partition, equal characters & sorted large partition to obtain final sorted string.

Ex: string : openai

lets say 'p'

- Partitioning : smaller = (enoai)

larger = (o)

Equal = (p)

Recursively sort larger & smaller partitions.

Sort "enoai".

sort "o"

Concatenate sorted smaller (aeinoo) with equal part (p) & sorted larger partition (" ")

The final sorted string is "aeinoop"

The runtime of algorithm is O(n) because each step we divide string into 3 parts (smaller larger & equal) & process each part separately.

Since pivot choosen Randomly we expect size of each partition is proportional to n. ensuring that the Algorithm runs in linear time.