SRIKAR KALE
2020701018

Subject: END QUIZ (Date / / (التاريخ) الموضوع:

Q1) $\phi = C_1 \wedge C_2 \wedge C_3 \wedge C_4$

$C_1 = x_2 \vee \neg x_3 \vee x_4$

$C_2 = x_1 \vee x_2 \vee \neg x_4$

$C_3 = x_2 \vee x_4$

$C_4 = x_3 \vee \neg x_2$

ILP:

let y_i represent x_i, then z_i represents C_i

$C_1 : x_2 \vee \neg x_3 \vee x_4 \longrightarrow y_2 + (1 - y_3) + y_4 \geq z_1$

$C_2 : x_1 \vee x_2 \vee \neg x_4 \longrightarrow y_1 + y_2 + (1 - y_4) \geq z_2$

$C_3 : x_2 \vee x_4 \longrightarrow y_2 + y_4 \geq z_3$

$C_4 : x_3 \vee \neg x_2 \longrightarrow y_3 + (1 - y_2) \geq z_4$

Where $y_1, y_2, y_3, y_4, z_1, z_2, z_3, z_4 \in \{0, 1\}$

Subject: (Date / / التاريخ:) الموضوع:

Q3) Work complexity of a parallel algo? When is the parallel algo said to be optimal?

Work complexity talks about the amount of work done by all the processors combined to get the required out put.

Let us assume we used 'p' processors to a particular problem and achieved a speedup, which is given by $T_s(A, p)$

$A \rightarrow$ problem / problem instance
$p \rightarrow$ No of processors used.

$$W(A) = p \cdot T_s(A, p)$$

We say that a parallel algorithm is optimal when

$$\underbrace{T_s(A, P)}_{\text{Time taken in parallel}} < \underbrace{T_s(A, 1)}_{\text{Time taken in sequential algorithm}}$$

&

$$W(A) = p \cdot T_s(A, P) \le T(A, 1) \quad (\text{Work done in seq} = 1 \cdot T(A, 1))$$

Work done in parallel

i) Time taken by the parallel algo should be lesser than the time taken by the sequential algorithm

ii) Work done by the parallel algo should be lesser than or equal to the time/work done by sequential algorithm.

Subject: ______ (Date / / التاريخ:) ______ الموضوع:

Q4) Consider n people picking a number between 1 & n (both inclusive) independently & uniformly at random. Find the expected no of people who pick the number 1. Find the expected no of people who pick the same no?

$X = \sum_{i=1}^{n} X_i$ X_i is a random variable

X_i is defined as (Indicator random variable) $\begin{cases} 1 & \text{if } i^{th} \text{ choice} = 1 \\ 0 & \text{otherwise.} \end{cases}$

$Pr(X_i = 1) = \frac{1}{n}$ } one choice in n

$E[X_i] = \sum i \cdot Pr(X_i = i)$

$= 1 \cdot Pr(X_i = 1) + 0 \cdot Pr(X_i = 2, 3, \ldots)$

$= \frac{1}{n}$

$E[X] = n \times \frac{1}{n} = 1$

Q) Define G to be an r-partite graph. Design a randomized algorithm that obtains a subgraph H of G such that H is r-partite & H has as many edges of G as possible. What is the lower bound on the no of edges of H acc to your algorithm.

Let H be an r-partite graph. Let the no of vertices in G be 'n'.

Every vertex chosen u.a.r and independently has a chance of going to one of the 'r' partites of H.

$$P_r(V_i) = \frac{1}{r}.$$

So an edge (U, V) is possible when $U \neq V$.

No of choices of $u = r$, no of choices of $v = r$

No of choices of '$U=V$' $= r$

$\therefore$ No of edges possible will be $\underbrace{r^2}_{\text{Total edges combination}} - \underbrace{r}_{\text{similar } U \& V}$

Total no of edges will be r^2

$$\therefore |E(H)| \leq |E(G)| \cdot \frac{r^2 - r}{r^2} \leq \frac{r(r-1)}{r^2} |E(G)|$$

$$\leq \left(\frac{r-1}{r}\right)|E(G)|$$

Subject: (Date / / :التاريخ) :الموضوع

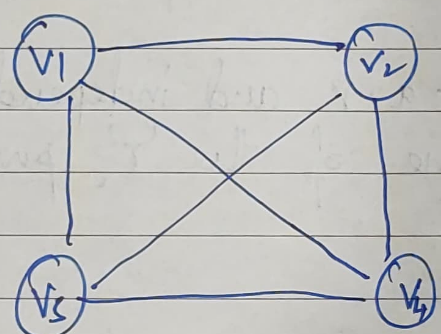
Ex: $r=2$

$$|E(H)| \le \frac{|E(G)|}{2}$$

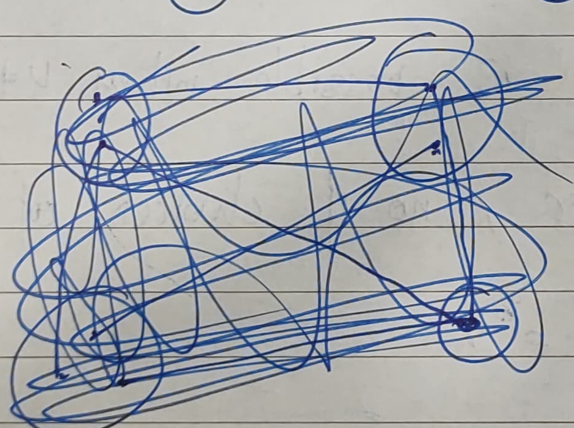
$r=3$

$$|E(H)| \le \frac{2}{3}|E(G)|$$

Lower bound will be ${}^{r}C_2$ edges (one edge from each partition)



$r=4 \rightarrow 6$ vertices minimum



Section 2

Subject: (Date / / التاريخ:) الموضوع:

What is accelerated cascading?

A → $O(\log n)$ time
$O(n \log n)$ work ← fast & ~~suboptimal~~

B → $O\left(\log n / \log\log n\right)$ time
$O(n)$ work. ← slow & optimal

C → $O(\log n)$ time
$O(n)$ work. } Accelerated cascading algo.

We have one ~~[illegible]~~ algorithm which is fast but is suboptimal in work & we have one algo which is slow but optimal in work. In accelerated cascadin[g] we utilize both these algo's in a manner which gives us a fast as well as an optimal algorithm.

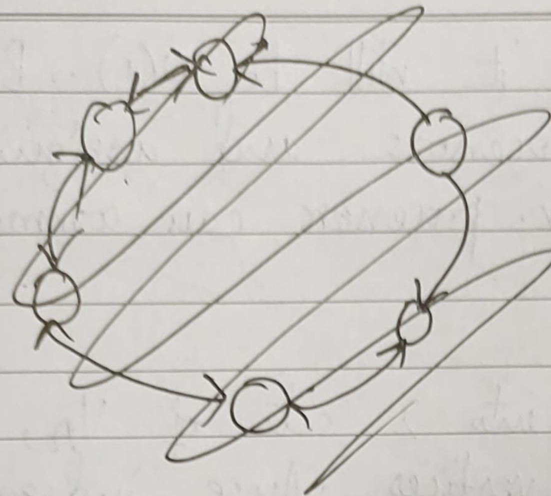
Start with slow but optimal ~~[illegible]~~ Algo till problem is small enough

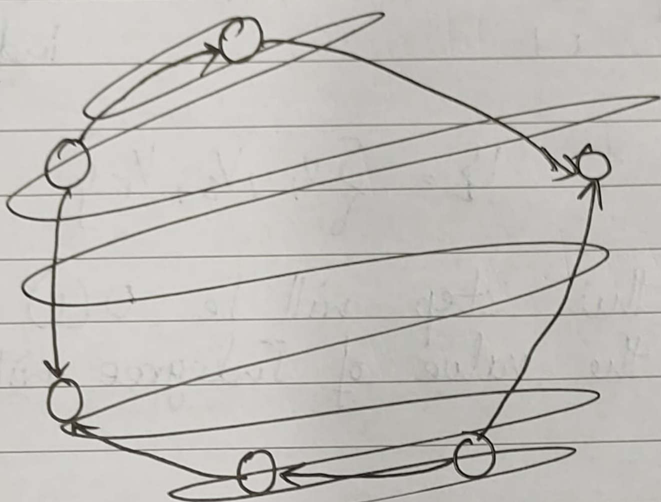
Switch over to the fast algo for the remaining.

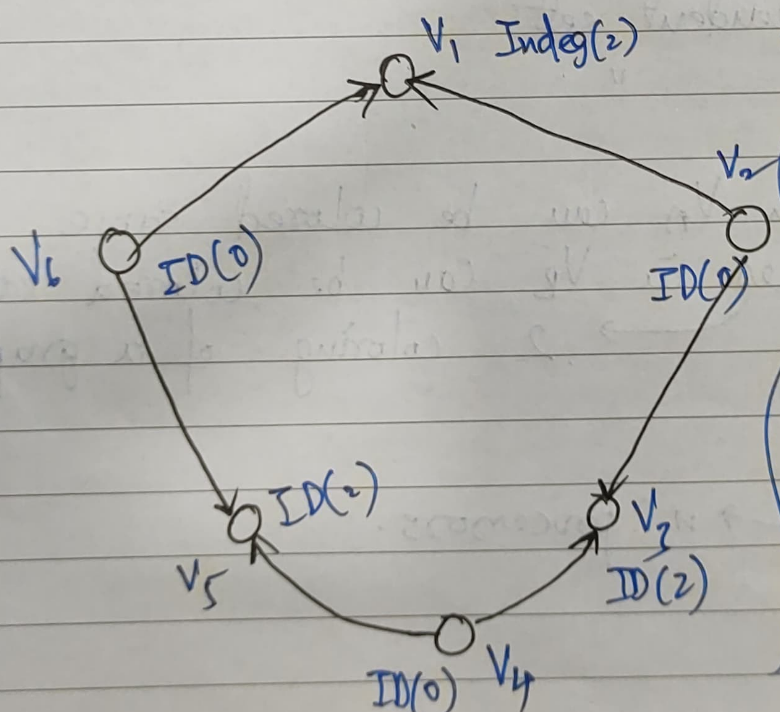
Start with B, reduce i/p size then switch to A

Subject: ______ (Date / / التاريخ:) ______ الموضوع:

Q)







→ No of nodes in such a graph is even.

→ We can segregate the nodes into 2 groups based on either their indegree or out degree

→ Assume their indegrees are not known.

Then time taken to find it will be $O(1)$. Because we are taking 'n' processors and assigning each processor a vertex. processors can communicate with their neighbors.

Step 2:

Divide the vertices into 2 sets of V_A, V_B

V_A is the set of all vertices whose indegree = 0

V_B " " indegree ≠ 0.

$V_A = \{V_2, V_4, V_6\}$ $V_B = \{V_1, V_3, V_5\}$

Time taken in this step will be $O(1)$ as the processors can store the value of Indegree with them.

Step 3:

V_A is an independent set

V_B " " "

So all vertices in V_A can be colored same similarly all vertices in V_B can be colored same

→ 2 coloring of a graph.

Run time : $O(1)$

Workdone = $O(n)$

→ n processors.