Distributed IS (Part 1)

Outline

- Independent Set (IS)

- Distributed IS
  1. Distributed Slow MIS
  2. Distributed Fast MIS

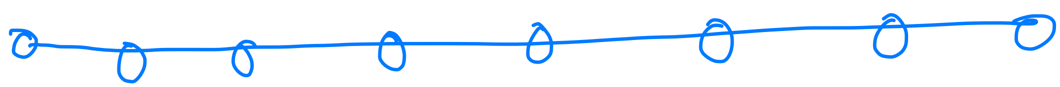
Independent Sets

Inependent Sets

- **IS:** Given an undirected Graph $G = (V, E)$ an independent set is a subset of nodes $U \subseteq V$, such that no two nodes in U are adjacent.

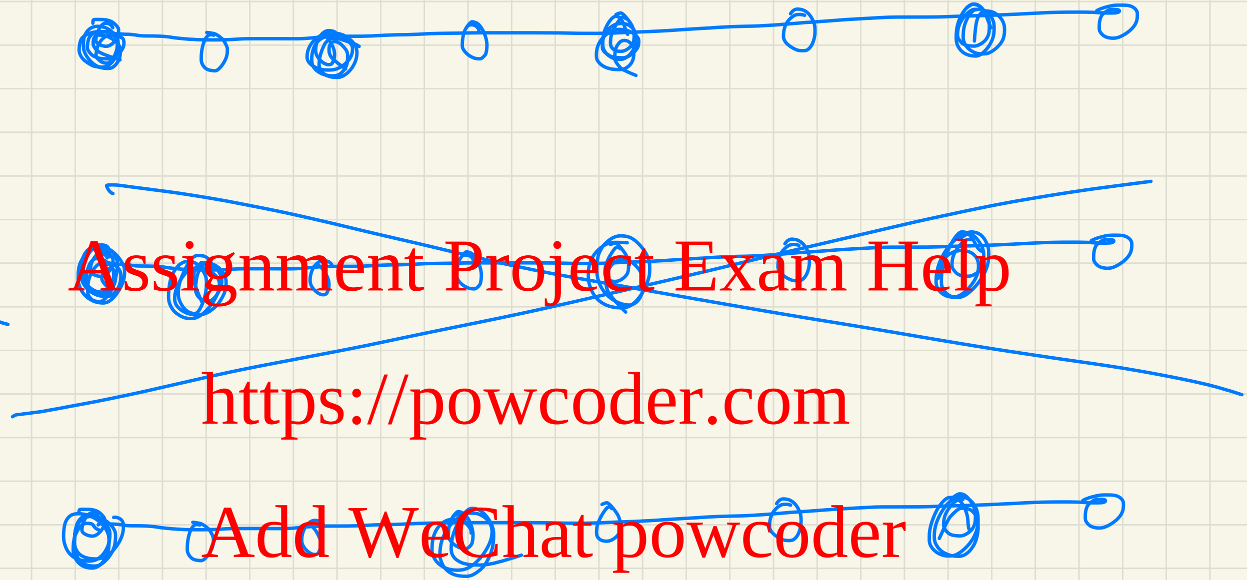
- **MIS:** An independent set is maximal if no node can be added without violating independence.

- **MaxIS:** An independent set of maximum cardinality is called maximum.



maximal of size 4

maximum



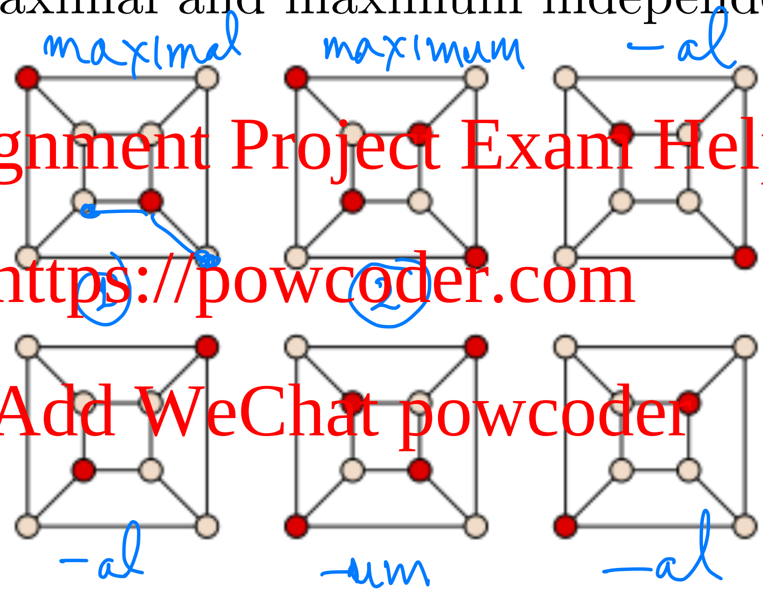
maximal of size 3

~~Does it have a maximum is of size 5?~~

Independent Set (IS)

- An IS is a set of nodes of the graph such that any two of them are not adjacent.

- We also have maximal and maximum independent sets.



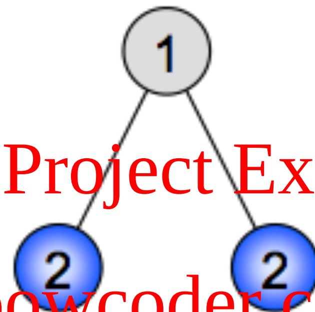
- Every MIS (Maximal Independent Set) is a dominating set.
- In general, the size of every MIS can be larger than the size of an optimal minimum dominating set by a factor of $\Omega(n)$.[a]

[a] We won't prove this here.

A set D is a dominating set, if every of the graph is adjacent to some vertex in D

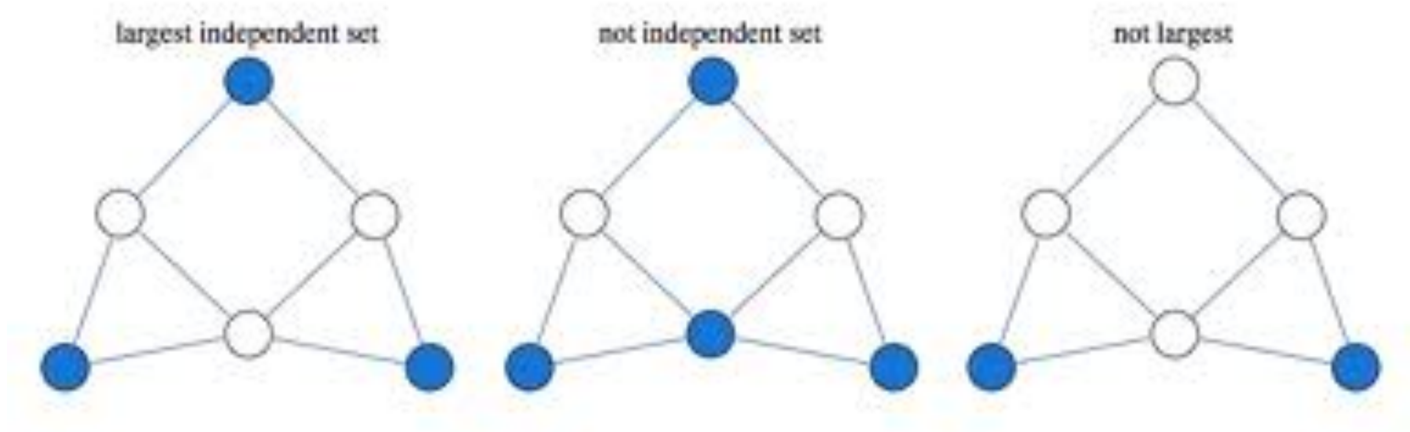
Coloring and Independent Sets

- **Example 1** Graph has two maximal independent sets (MIS), but only one is a maximum independent set (MaxIS).



In a colored graph, the vertices of a color form an IS.

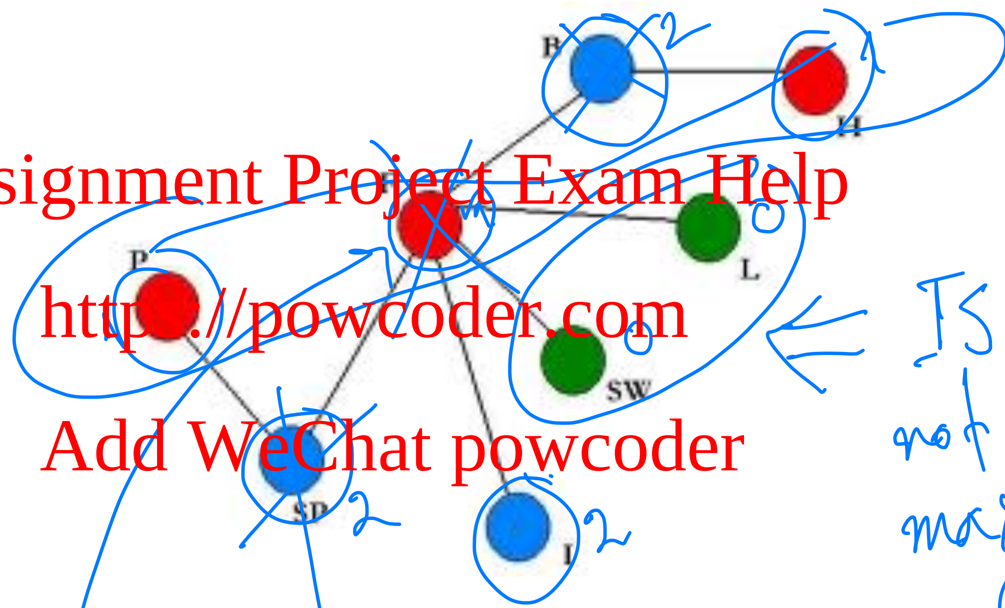
- **Example 2**



Coloring and Independent Sets

- There is a relation between independent sets and node coloring:
  - each color class is an independent set,

IS 3
is maximal



IS
not maximal
2

  - however, it is not necessarily a MIS.

Joyce Blue + Green : Max IS
Me : Tree with root : it has 5 leaves

From Coloring to Independent Sets

- Starting with a coloring, one can derive a MIS algorithm:

1. We first choose all nodes of the first color.
2. Then, for each additional color we add "in parallel" (without conflict) as many nodes as possible.

Assignment Project Exam Help

Choose a color = 0

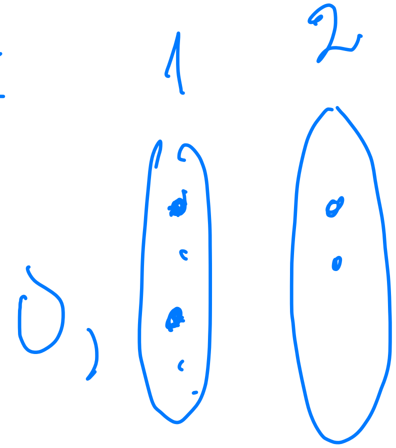
https://powcoder.com

Add WeChat powcoder

$\underbrace{1, 2, \dots, c-1}$

parallel

$X = c$



From Coloring to Independent Sets: Analysis

- **Theorem 1** *Given a coloring algorithm that needs C colors and runs in time T, we can construct a MIS in time $C + T$.*

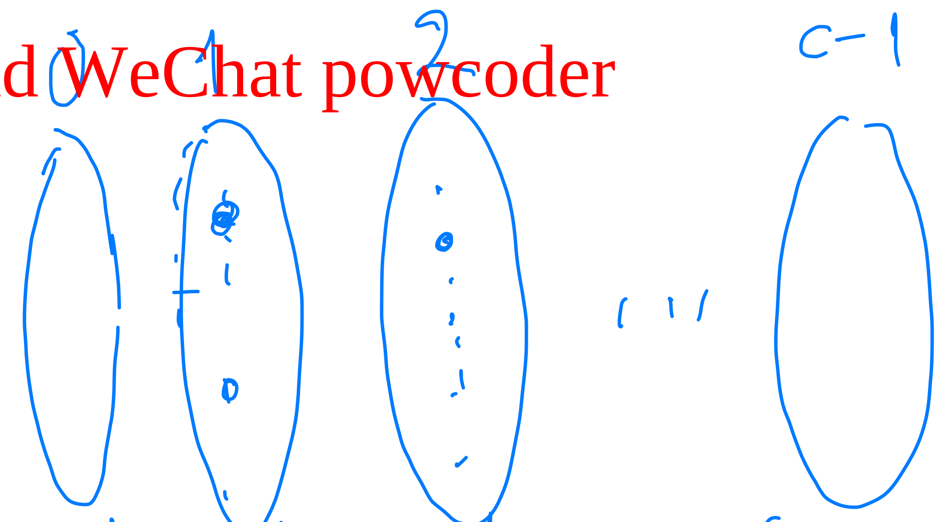
- Time complexity:
  - the T in the time complexity comes from the coloring algorithm, and
  - the C in the time complexity comes from the number of colors.

X_i = nodes of color i

$X_0 \cdot X_1 +$

$(X_0 + X_1)\, X_2$

$\vdots$



Time Steps 1st 2nd 3rd (c-1)-step

Time Complexity is $C+T$

Message Complexity:

X_i = # of nodes of color i

$$X_0 X_1 + (X_0 + X_1) X_2 + (X_0 + X_1 + X_2) X_3$$

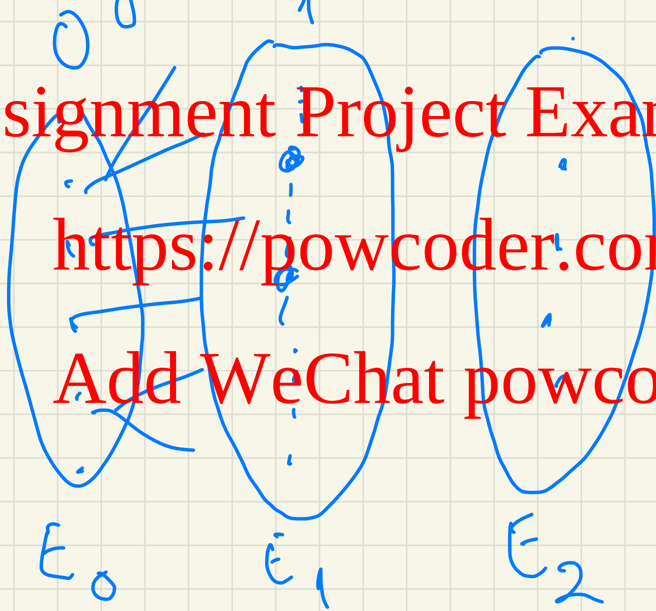
$$\underbrace{\ldots + \underbrace{(X_0 + \cdots + X_{c-2})}\, X_c}_{n^2}$$

Message Complexity is in $O(c \cdot n^2)$, $c = \chi$ = chromatic number

In addition we need the cost of the coloring alg.

$$\left\{\begin{matrix}\text{Message} \\ \text{Complexity} \\ \text{of FS} \\ \text{algorith}\end{matrix}\right\} \le \left\{\begin{matrix}\text{Message} \\ \text{Compl.} \\ \text{of Coloring} \\ \text{Algorith}\end{matrix}\right\} + \left\{\begin{matrix}\text{above} \\ \text{calculation}\end{matrix}\right\}$$

Joyce: Can't we do it in $O(E)$ messages? YES



Cost = Cost of Col. + $O(E)$

It needs: $\underbrace{E_0 \cup E_1 \cup \dots}$ $\subseteq E$

Related Topic: Set Cover (SC)

- Given a set of elements $\{1, 2, \ldots, n\}$ (called the universe) and a collection S of m sets whose union equals the universe, the set cover problem is to identify the smallest sub-collection of S whose union equals the universe.

  $S = \{S_1, \ldots, S_m\}$

  1. For example, consider the universe $U = \{1, 2, 3, 4, 5\}$ and the collection of sets $S = \{\{1, 2, 3\}, \{2, 4\}, \{3,4\}, \{4,5\}\}$. Clearly the union of S is U. However, we can cover all of the elements with the following, smaller number of sets: $\{\{1, 2, 3\}, \{4, 5\}\}$.
  2. A company needs to buy a certain amount of varied supplies and there are suppliers that offer various deals for different combinations of materials (Supplier A: 2 tons of steel + 500 tiles for \$$x$; Supplier B: 1 ton of steel + 2000 tiles for \$$y$; etc.). You could use set covering to find the best way to get all the materials while minimizing cost.

Issues

- Computing a maximum independent set (MaxIS) is a notoriously difficult problem.

  - Equivalent to maximum clique on the complementary graph.
  - Both problems are NP-hard, in fact not approximable within $n^{1/2-\epsilon}$.

- MIS and MaxIS can have very different sizes.

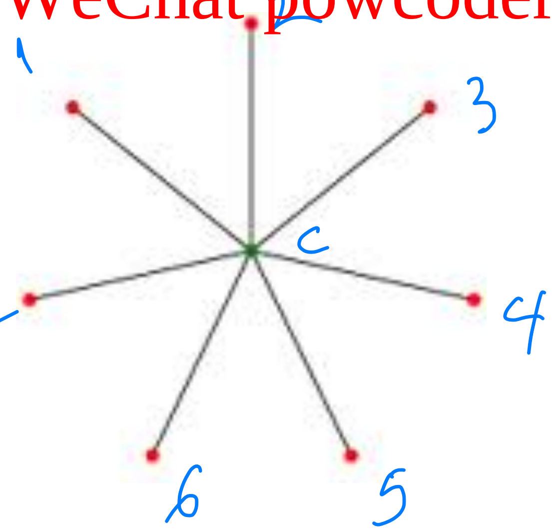
  - On a star graph MIS is $\Theta(n)$ smaller than the MaxIS.

$\{c\}$ is MIS

$\{1, 2, .., 7\}$ MaxIS

MIS : size 1

MaxIS : size n-1



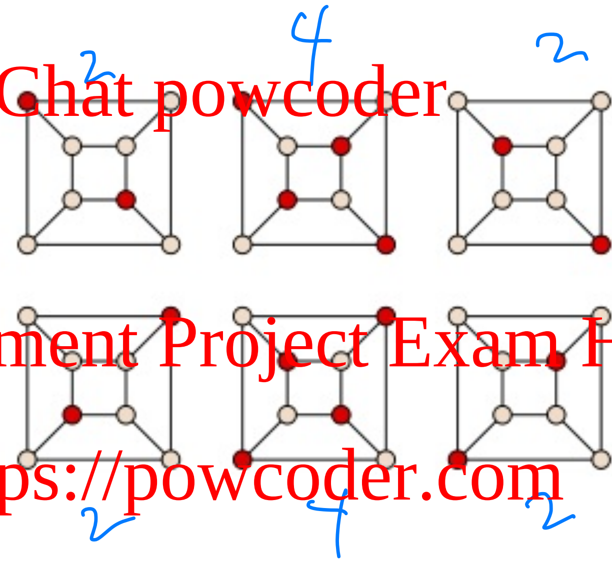
n leaves

1 central node

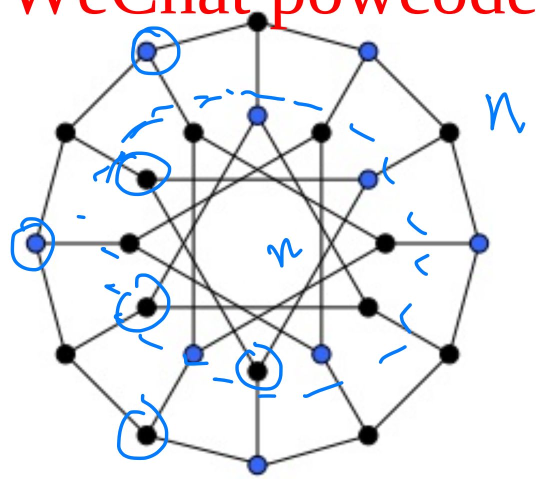
Star with n nodes

Examples: MIS

- Example 1



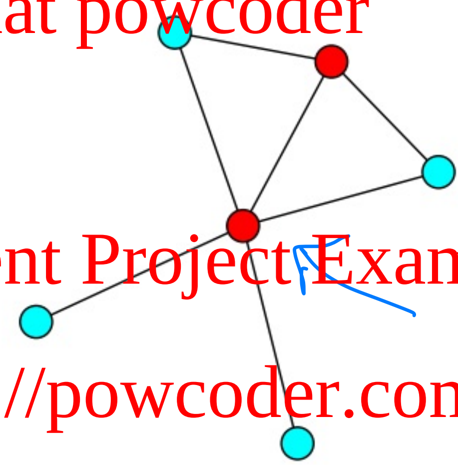
- Example 2



MaxIS has size n

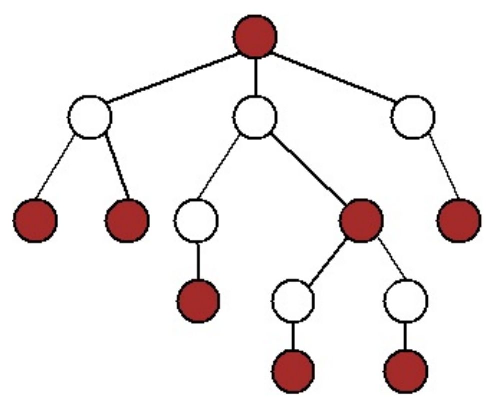
Examples: MIS

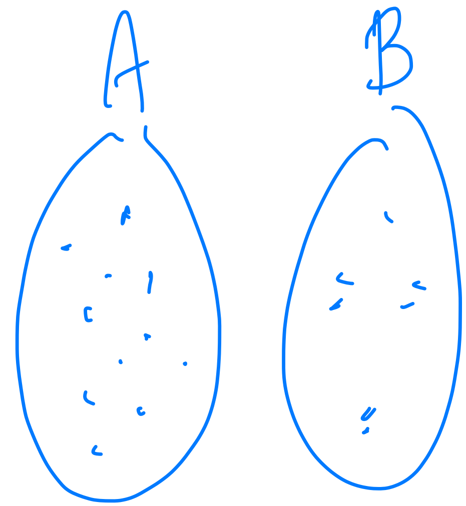
- Example 3



MIS

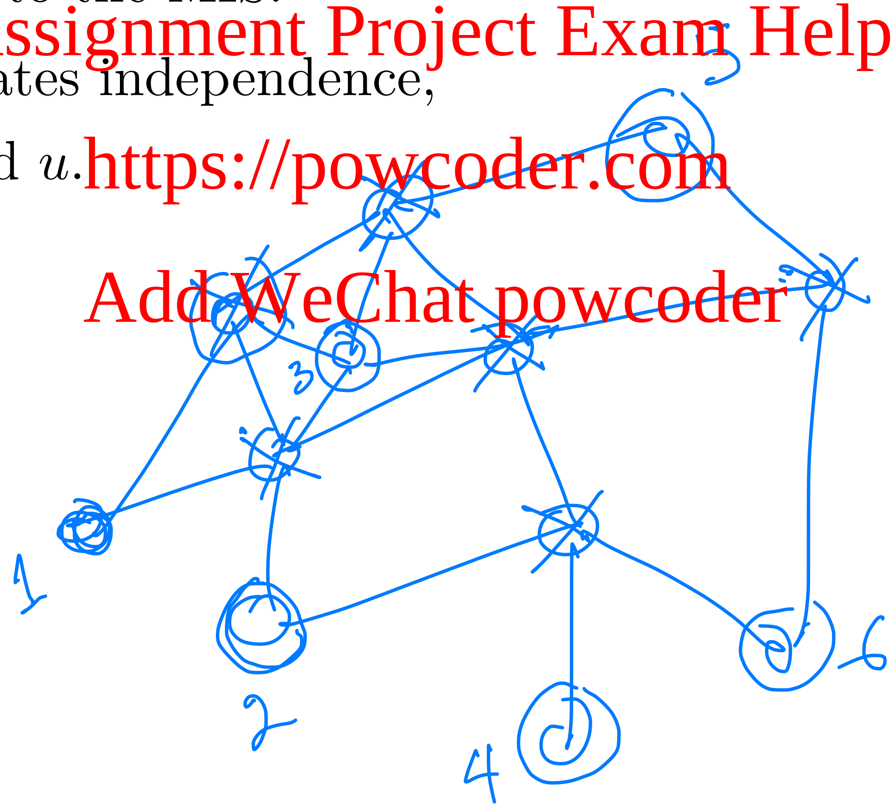
- Example 4



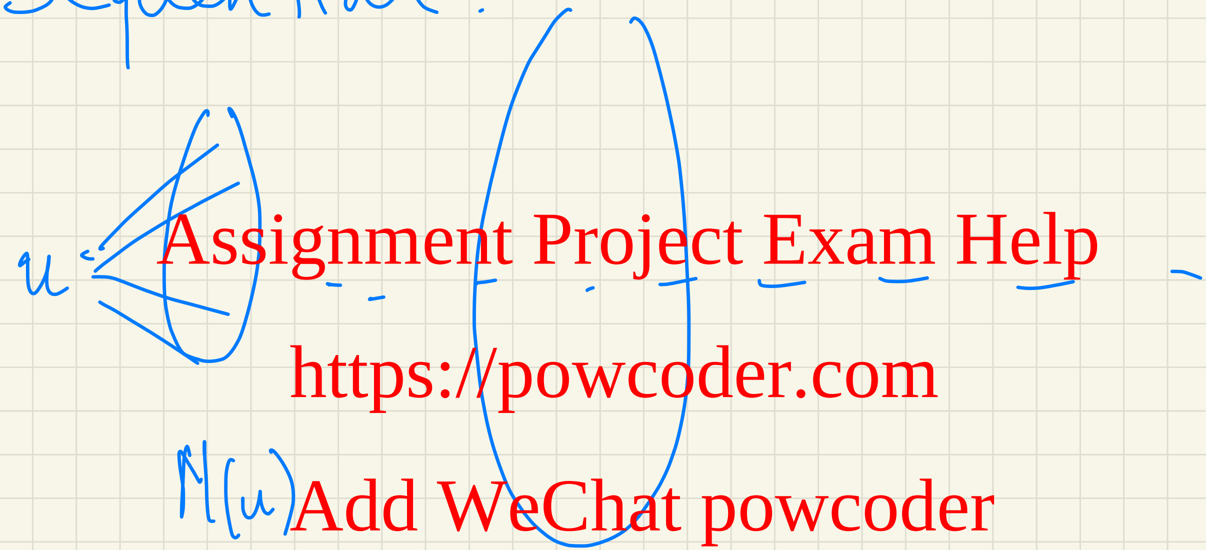


Computing MIS

- Computing a MIS sequentially is trivial:
  1. Scan the nodes in arbitrary order.
  2. If a node u does not violate independence,
     – add u to the MIS.
  3. If u violates independence,
     – discard u.



Sequential:



$N(N(u))$

$N_2(u)$

Algorithm: Lexicographic $MIS(G)$

- Previous algorithm sometimes stated as follows. Consider a graph $G = (V, E)$ in which the vertices are lexicographically ordered.

$I = \emptyset, V' = V$
while $V' \neq \emptyset$ **do**
    Choose $v \in V'$ (in lexicographic order)
    $I \leftarrow I \cup \{v\}$
    $V' \leftarrow V' \setminus (\{v\} \cup N(v))$
Return I;

- With this simple greedy algorithm, we can find a MIS in $O(|V| + |E|)$ time.
- The main question is how to compute a MIS in a distributed manner.

Distributed

Distributed Slow MIS

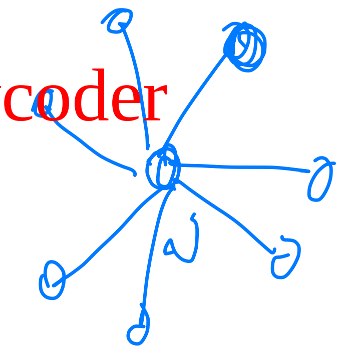
- Main idea is to give priority to nodes with higher ID.

- **Slow MIS**
- Requires Node IDs
- Every node v executes the following code:

1. **if** all neighbors of v with larger identifiers have decided not to join the MIS then

2. v decides to join the MIS

3. **end if**



v queries its neighborset N(v)

Complexity

- **Theorem 2** *Algorithm Slow MIS has time complexity of* $O(n)$ *and a message complexity of* $O(m)$.

- Slow MIS is not better than the sequential algorithm in the worst case, because there might be one single point of activity at any time.

Issues

we will discuss this later

- Using Theorems 1 and 2 we get a distributed deterministic MIS algorithm for cycles (also for trees, and bounded degree graphs) with time complexity $O(\log^* n)$ (will cover this later in class).
  - First do the colouring in $O(\log^* n)$ rounds.
  - Choose all nodes of the first color.
  - For each additional color we add in parallel (without conflict) as many nodes as possible.

$\log \cdots \log\log$

$2^{2^{2^{2^{2^{2^{2^{2^{2}}}}}}}}$

$\log^* n = 9$

- With a lower bound argument one can show that this deterministic MIS algorithm for rings is asymptotically optimal.
  - Because in the ring MIS is "essentially" the same as coloring.
- There have been attempts to extend the 6-Color Algorithm to more general graphs, however, so far without much success.

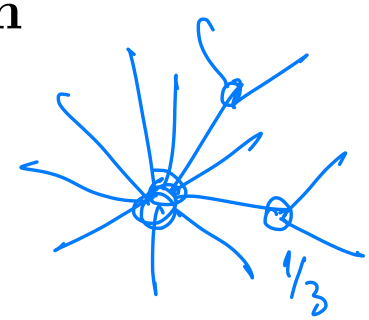
Is There a Faster Algorithm?

- Given that "Slow MIS" is not better than the sequential algorithm in the worst case
  - Is there a faster MIS?

- In the sequel we give a probabilistic algorithm with $O(\log n)$ expected termination time.

Goal: Find a parallel MIS algorithm

selected w p $\frac{1}{10}$



- Consider algorithms of the form

1. $I = \emptyset, V' = V$
2. While G' is not the empty graph
   (a) Choose a random set of vertices $S \subseteq V$ by selecting each vertex v independently with probability $\frac{1}{d_v}$, where d_v is the degree of v.
   (b) For every edge $(u, v) \in E(G')$ if both endpoints are in S, then remove the vertex of lower degree from S (break ties). Denote the set after this step as S'.
   (c) Remove S' and $Neighbor(S')$ and all adjacent edges from G'.
   (d) $I \leftarrow I \cup S'$

Distributed Fast MIS[a]

- Algorithm operates in synchronous rounds, grouped in phases.

- A single phase is as follows:

not required

1. Each node v marks itself with probability $\frac{1}{2d(v)}$, where $d(v)$ is the current degree of v.

2(a) If no higher degree neighbor of v is also marked, node v joins the MIS. /* Priority to nodes of higher degree */

(b) If a higher degree neighbor of v is marked, node v unmarks itself again.

/* If neighbors have same degree, ties broken by ID */

3. Delete all nodes that joined the MIS and their neighbors /* as they cannot join the MIS anymore. */

[a] A more general form of this algorithm assigns real numbers (in the range $[0, 1]$) as weights at the nodes. An alternative version is to label the vertices with a random permutation.

Issues

- Correctness in the sense that the algorithm produces an independent set is relatively simple.

  - Steps 1 and 2 make sure that if a node v joins the MIS, then v's neighbors do not join the MIS at the same time.
  - Step 3 makes sure that v's neighbors will never join the MIS.

- The algorithm eventually produces a MIS, because the node with the highest degree will mark itself at some point in Step 1.

- The only remaining question is how fast the algorithm terminates.
  - This is not easy to figure out!

Exercises[a]

1. Show that any maximal matching is a 2-approximation of a maximum matching.

2. Let $G = (V, E)$ be the graph for which we want to construct the matching. Define the auxiliary graph G' as follows:

   - for every edge in G there is a node in G';
   - two nodes in G' are connected by an edge if their respective edges in G are adjacent.

   Show that a (maximal) independent set in G' is a (maximal) matching in G, and vice versa.

[a] Do not submit!