

<https://eduassistpro.github.io/>

Assignment Project Exam Help

Add WeChat `edu_assist_pro`

Assignment Project Exam Help

d

<https://eduassistpro.github.io/>

**Conn**e

Add WeChat `edu_assist_pro`

<https://eduassistpro.github.io/>

## Assignment Project Exam Help

- Trees
  - Canonical Form
  - Distributed Views
  - Broadcast
  - BFS/DFS
  - Flooding
  - Convergecast
  - Applications
- Add WeChat edu\_assist\_pro
- <https://eduassistpro.github.io/>
- Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

Assignment Project Exam Help

Add WeChat edu\_assist\_pro

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

# Trees and Communication Assignment Project Exam Help

- Trees are everywhere: sapling                              al compounds.  
– There is something about t Add WeChat edu\_assist\_pro d economy.

# Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

- Trees form a natural communication structure in distributed computing.

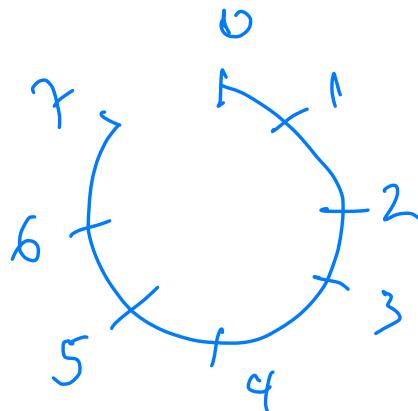
<https://eduassistpro.github.io/>

## Main Concepts on Trees Assignment Project Exam Help

- A tree is a connected graph that ha  
**Add WeChat edu\_assist\_pro**
- Start with the tree of one vertex: w  
ree we  
wish by successively adding a new edge and a new vertex.
  - At each stage, the # of vertices exceeds the number of edges by 1, so every tree with  $n$  vertices has exactly  $n - 1$  edges  
**Assignment Project Exam Help**

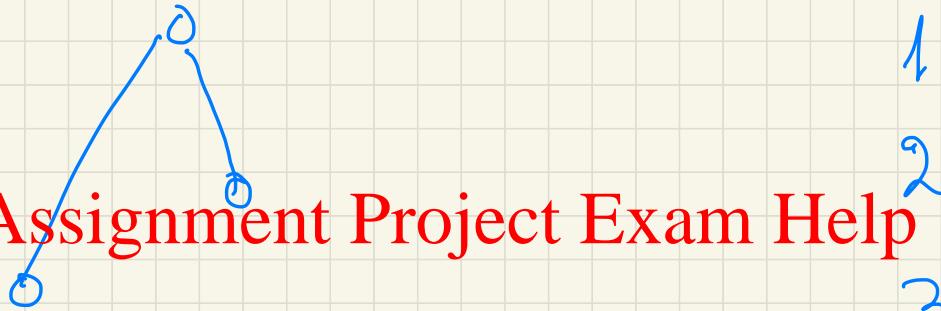
<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



$$\begin{aligned} & u_0, u_1, \dots, u_{n-1} \\ & E(u_i, u_{i+1}), \quad 0 \leq i \leq n-1 \end{aligned}$$

Assignment Project Exam Help  
<https://eduassistpro.github.io/>  
Add WeChat edu\_assist\_pro



# Ver	# Edge
1	0
2	1
3	2

<https://eduassistpro.github.io/>

## Characterization of Trees Assignment Project Exam Help

- Let  $T$  be a graph with  $n$  vertices. The following statements are equivalent:

Add WeChat edu\_assist\_pro

- $T$  is connected and has no cycles.
- $T$  has  $n - 1$  edges and has no cycles.
- $T$  is connected and has  $n - 1$  edges.
- $T$  is connected and every vertex connects  $T$ .
- Any two vertices are connected by exactly one path.
- $T$  contains no cycles, but the addition of an edge creates a cycle.



Are equivalent

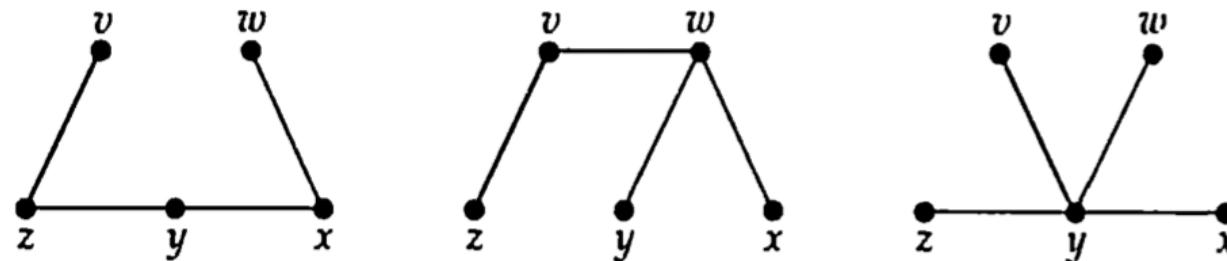
<https://eduassistpro.github.io/>

## Spanning Trees Assignment Project Exam Help

- Let  $G$  be a connected graph. A spanning tree in  $G$  is a subgraph of  $G$  that includes all vertices of  $G$  and is also a tree.
- A graph ...



- ... and possible spanning trees



- Spanning trees emerge naturally in communication.

<https://eduassistpro.github.io/>

## Assignment Project Exam Help

- A Forest is a collection of vertex di

Add WeChat [edu\\_assist\\_pro](#)

The trees  
of the forest  
form connected  
Component.

## Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat [edu\\_assist\\_pro](#)

- Forests arise naturally in clustering.
- A Spanning Forest is a collection of vertex disjoint spanning trees.

disjoint:  $\overline{T_1}, \overline{T_2}, \dots, \overline{T_k}$

Tree  $\overline{T_i}$  has  $t_i$  vertices  
Assignment Project Exam Help

$$f_L = \{ \overline{T_1}, \overline{T_2}, \dots, \overline{T_k} \}$$

<https://eduassistpro.github.io/>

How many WeChat add friends the forest have?

$n = \# \text{ of vertices in the forest}$

$$= t_1 + t_2 + \dots + t_k$$

$$t_1 - 1 + t_2 - 1 + \dots + t_k - 1 = n - k$$

<https://eduassistpro.github.io/>

Assignment Project Exam Help

Add WeChat edu\_assist\_pro

Assignment Project Exam Help

Ca<https://eduassistpro.github.io/>nn

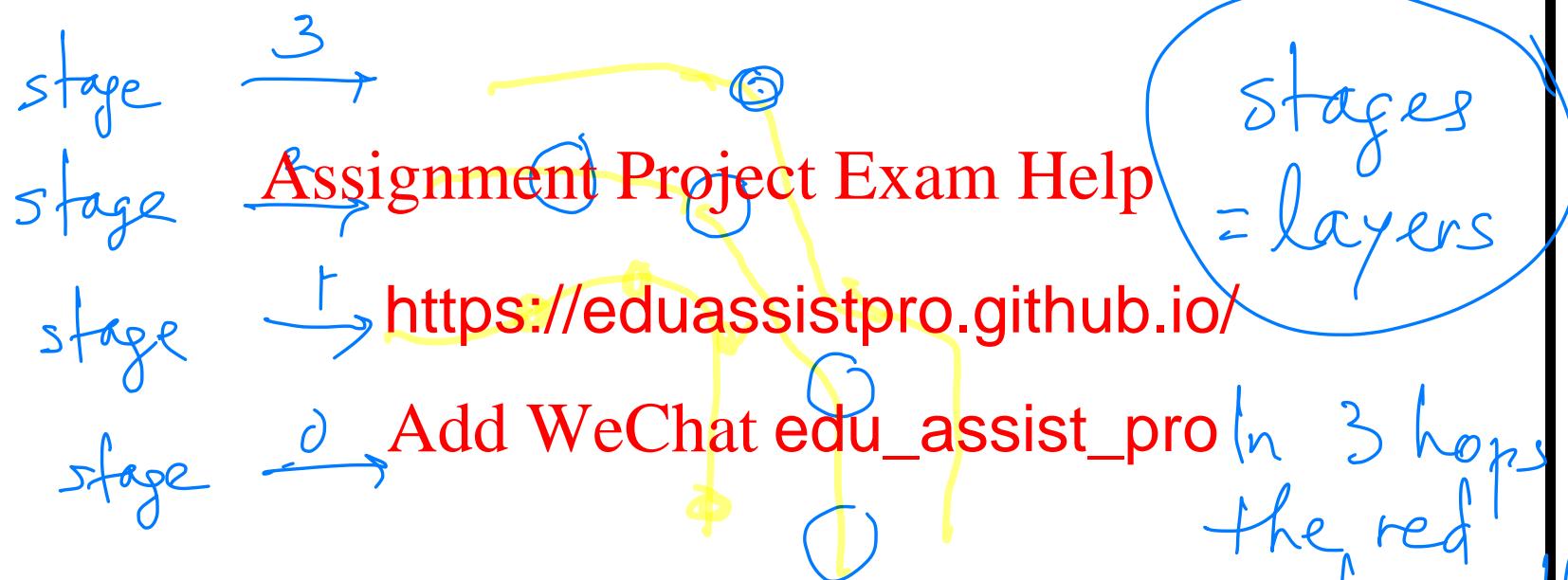
Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

## How Do Nodes Build their Knowledge? Assignment Project Exam Help

- They learn by exchanging mes

Add WeChat edu\_assist\_pro

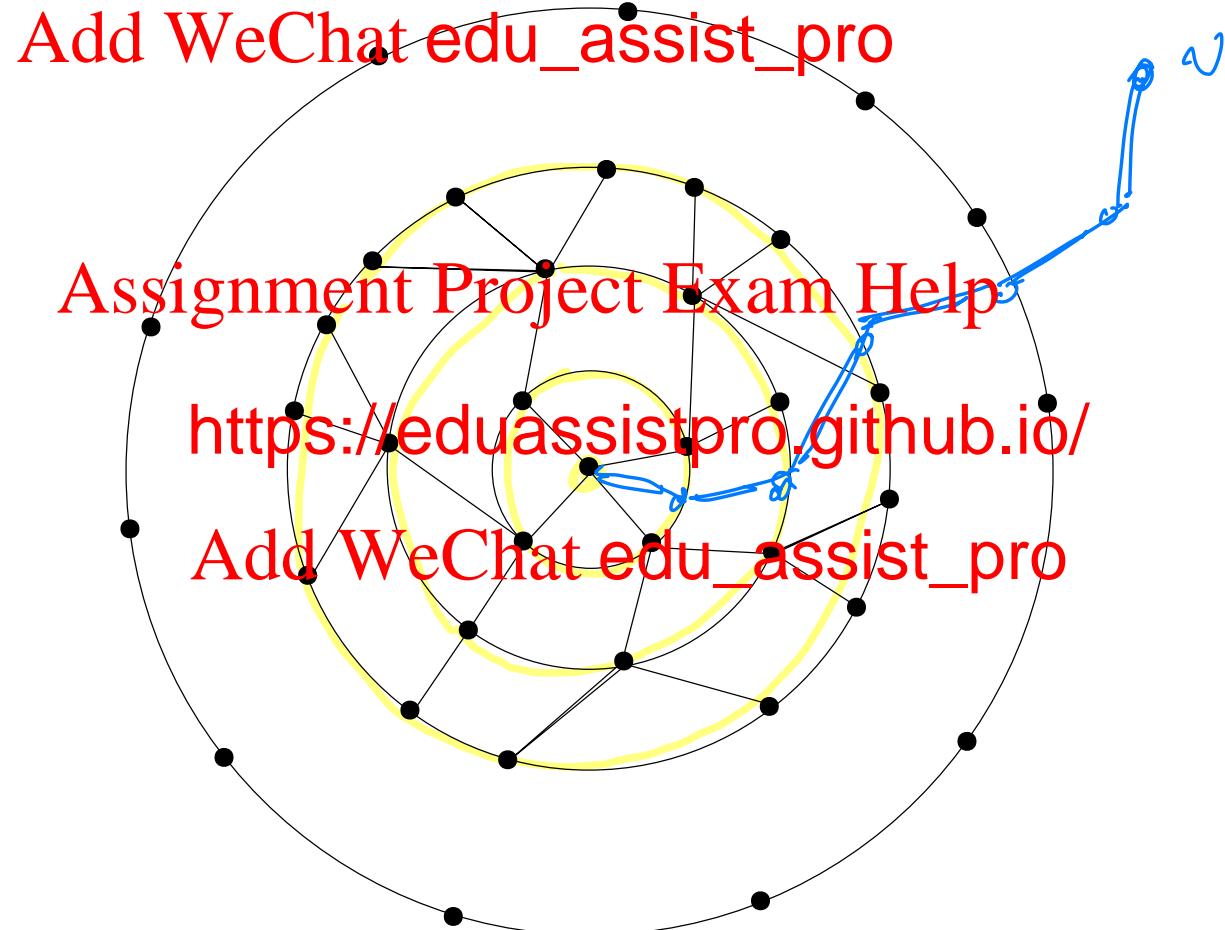


- At the same time, different nodes learn different things!

<https://eduassistpro.github.io/>

## Information Growth and Knowledge Discovery Assignment Project Exam Help

- Start from node  $u$



- When does the growth stop?

It stops when you reach  
the farthest you can

<https://eduassistpro.github.io/>

## In a Line Graph Assignment Project Exam Help

- In a typical synchronous distributed system each node executes the following atomic

Add WeChat edu\_assist\_pro  
send → receive → process

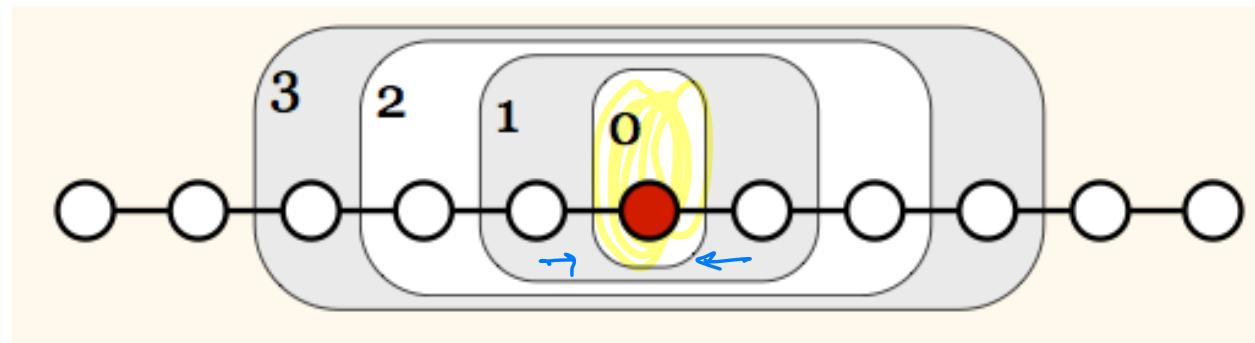
in synchronous rounds  
**Assignment Project Exam Help**

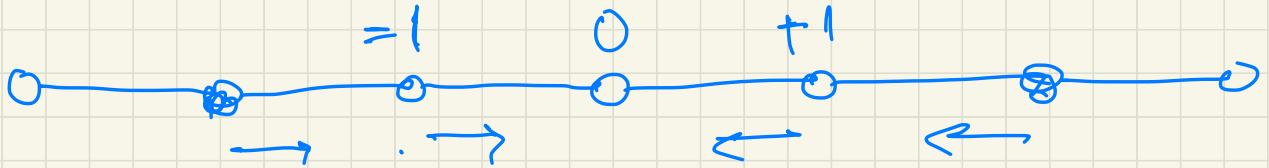
- Node  $v$ , by excha

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

- ...receives information about distance 1, 2, 3, ... nodes.





## Assignment Project Exam Help



Add WeChat edu\_assist\_pro



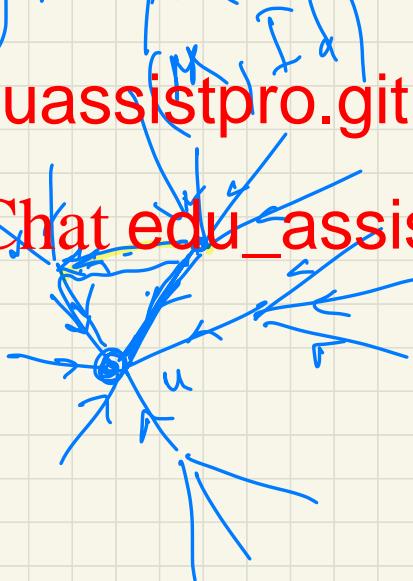
If builds knowledge about  
the graph.

The method is not limited  
by any graph: it works in any  
connected graph

Assignment Project Exam Help

<https://eduassistpro.github.io/>

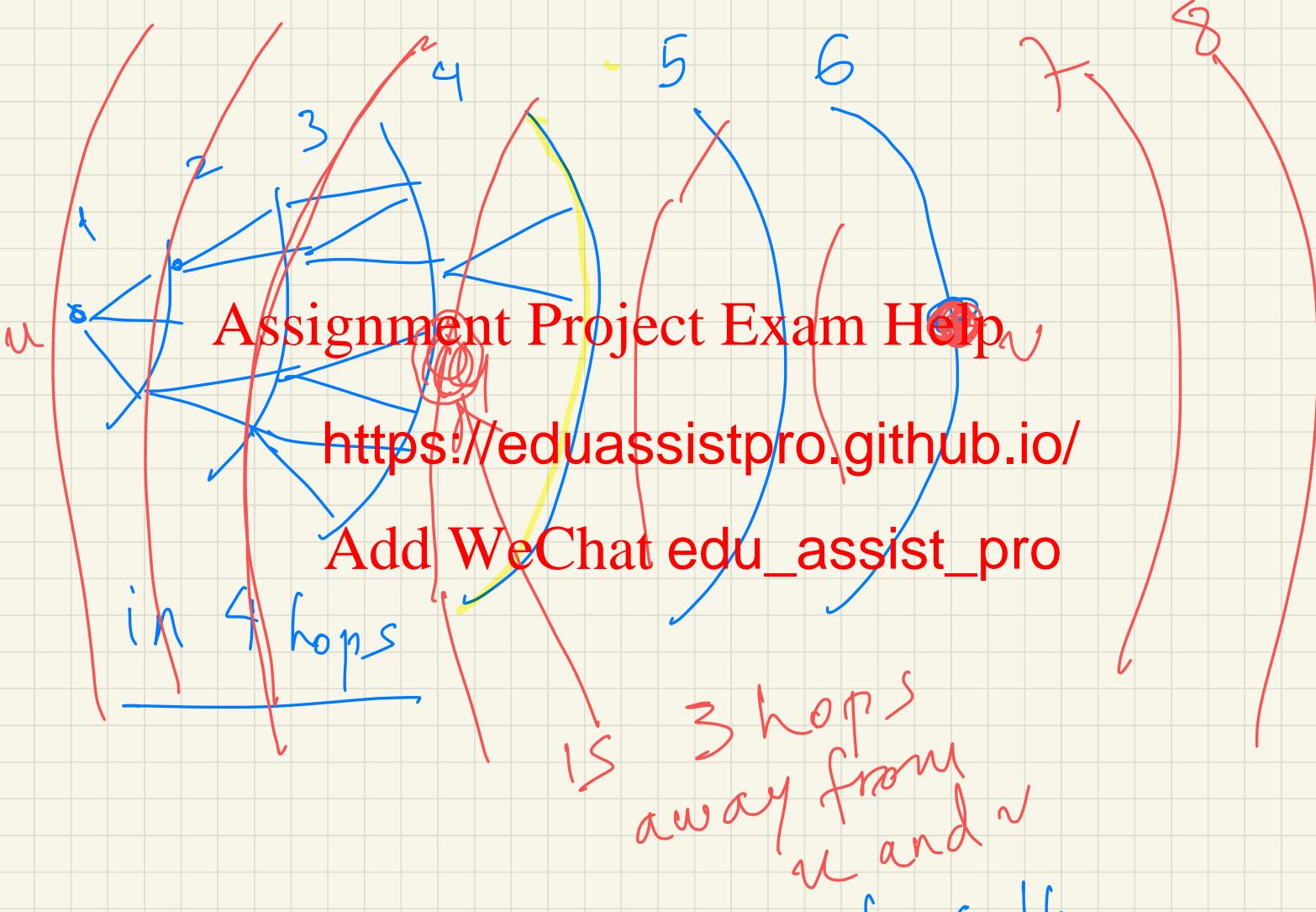
Add WeChat `edu_assist_pro`

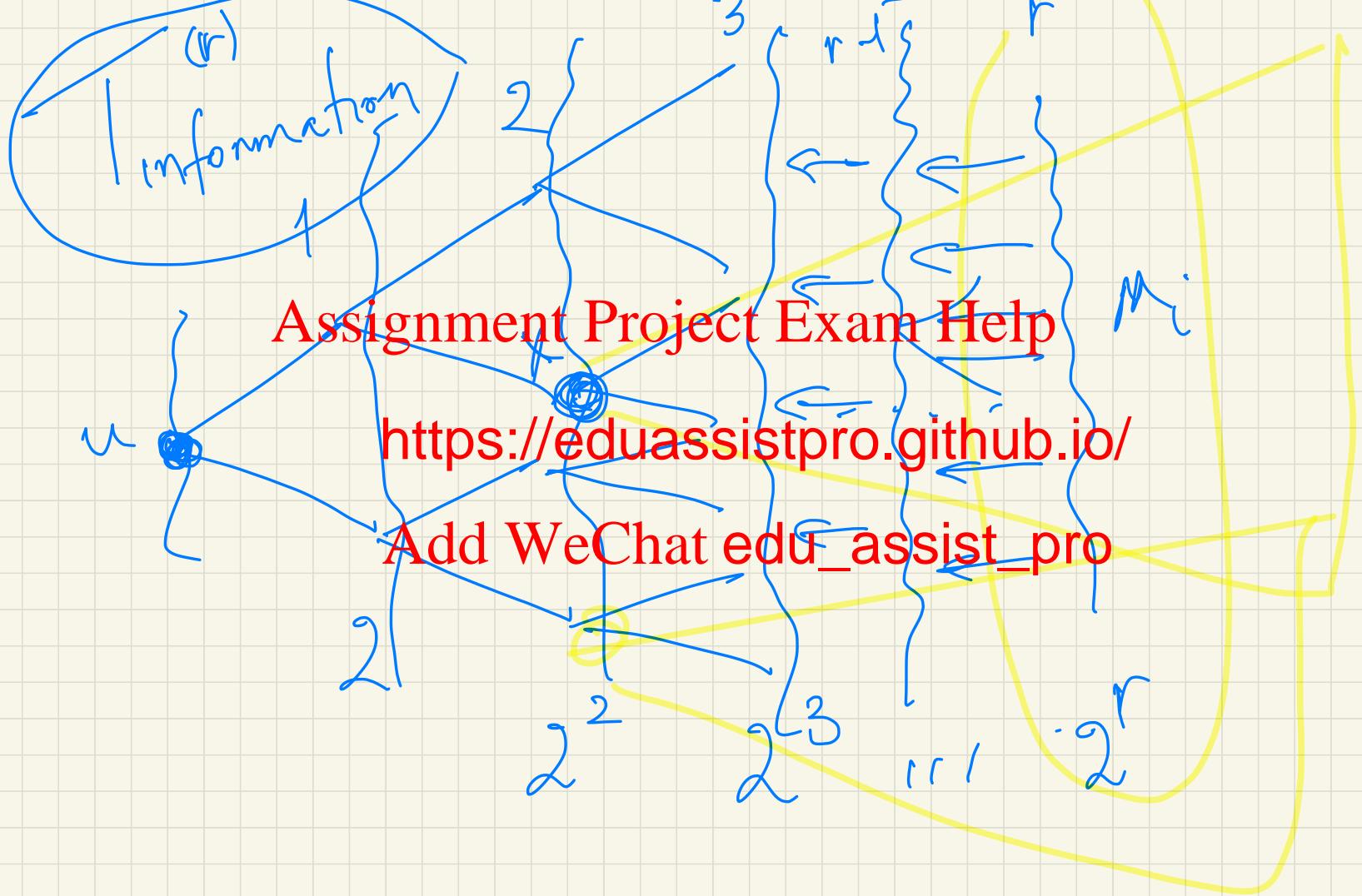


<https://eduassistpro.github.io/>

## Canonical Form Assignment Project Exam Help

- Assume that initially, all nodes are assigned unique identifiers (wn identifier).  
[Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/)
- Information needs at least  $r$  rounds to travel  $r$  hops.
- After  $r$  rounds, a node  $v$  can only learn about other nodes at distance at most  $r$ .  
[Assignment Project Exam Help](https://eduassistpro.github.io/)
- If message size and identifier length are bounded, it is in fact not hard to see that in  $r$  rounds, a node  $v$  can learn exactly all the node labels and inputs up to distance  $r$  from  $v$ .
- This allows us to transform every deterministic  $r$ -round synchronous algorithm into a simple canonical form.



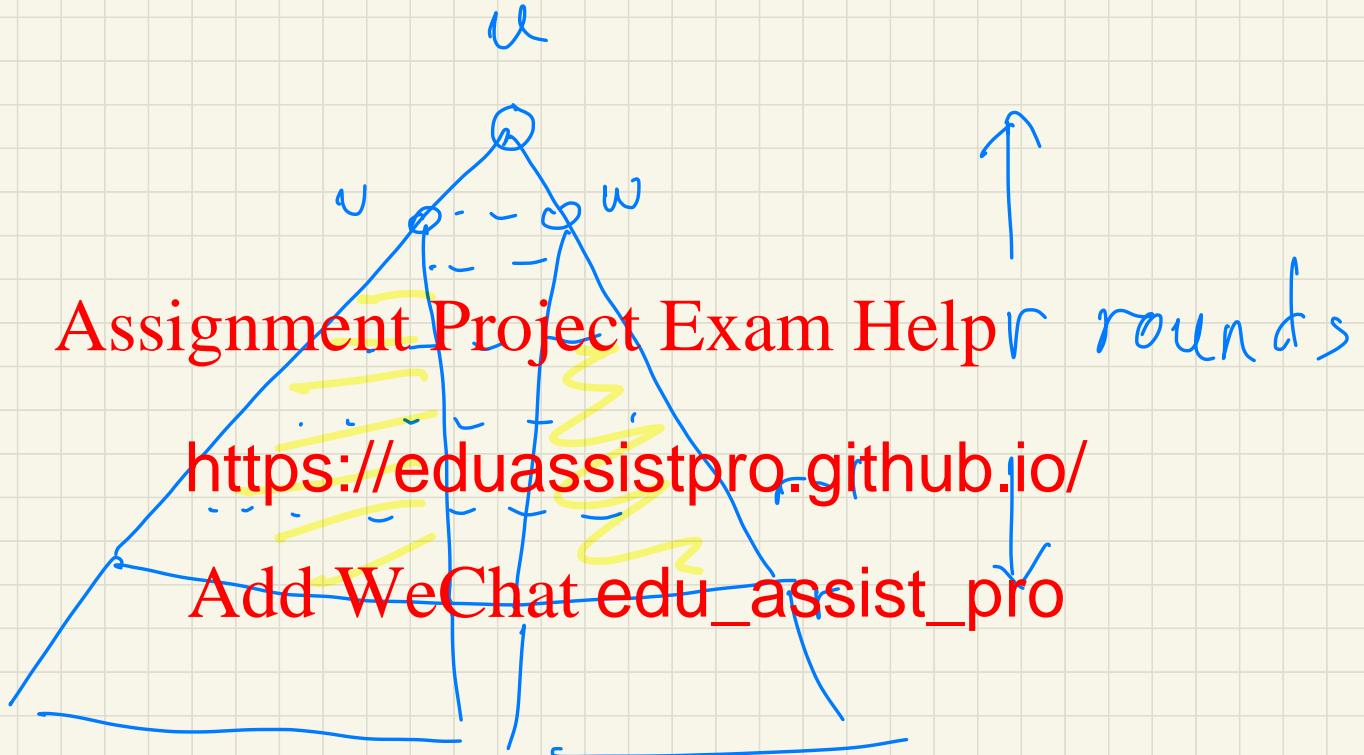


$2^0 + 2^1 + 2^2 + \dots + 2^{r-1} + 2^r$   
Assignment Project Exam Help

$https://eduassistpro.github.io/$

Add WeChat edu\_assist\_pro  
# of messages

An exponential number



<https://eduassistpro.github.io/>

## Cumulative Messages Assignment Project Exam Help

- The idea is to “simplify communication” by sending cumulative messages
- A typical synchronous distributed algorithm at each node consists of a sequence of executions

## Assignment Project Exam Help

ss

<https://eduassistpro.github.io/>  
in synchronous rounds

- Often what matters is the source and the destination
- Can we first do a sequence of  $r$  executions “send  $\rightarrow$  receive” followed by a single “process” at the end?
- In other words, can we send “cumulative” messages for  $r$  rounds and finally do the processing?

<https://eduassistpro.github.io/>

### Example: Computing the Sum in a Ring Assignment Project Exam Help

- Consider a ring of  $n$  nodes at each node, for  $i = 1, 2, \dots, n$ , [Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/) holds  $ID_i$  and weights  $w_i$

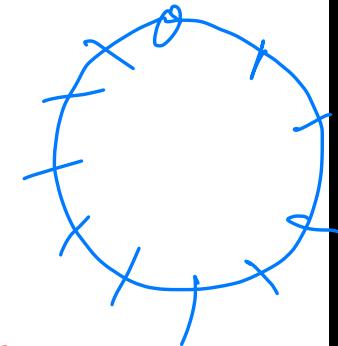
- In a typical distributed computation for a node  $i$ :

for  $r$  rounds do

- $i$  sends pair  $(ID_i, w_i)$  to  $i + 1$ , and receives pair  $(ID_{i-1}, w_{i-1})$  <https://eduassistpro.github.io/>
- process by adding  $w_i + w_{i-1}$

- This can be done in a cumulative manner at  $i$  as follows:

- $i$  sends  $(ID_i, w_i), (ID_{i-1}, w_{i-1}), \dots, (ID_{i-r}, w_{i-r})$  to  $i + 1$ , receives  $(ID_{i-1}, w_{i-1}), (ID_{i-2}, w_{i-2}), \dots, (ID_{i-r-1}, w_{i-r-1})$  from  $i - 1$
- process by adding  $w_{i-1} + w_{i-2} + \dots + w_{i-r-1}$



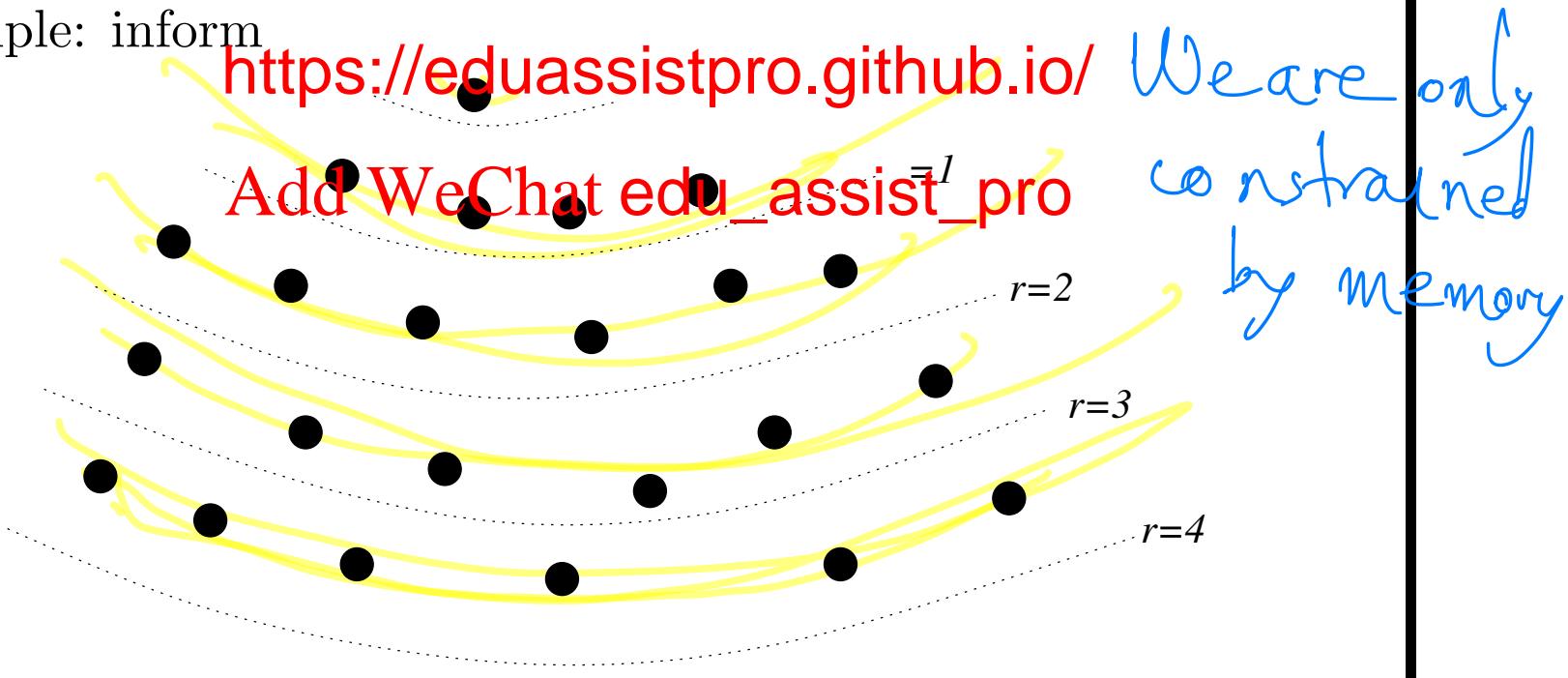
<https://eduassistpro.github.io/>

## Synchronous Algorithm: Canonical Form

### Assignment Project Exam Help

- Synchronous Algorithm
  - 1. In  $r$  rounds, send complete codes at distance at most  $r$  /\* all the communication first \*/
  - 2. Compute output based on complete information about  $r$ -neighborhood /\* do all the computation in the end \*/
- Example: inform

<https://eduassistpro.github.io/>



<https://eduassistpro.github.io/>

## Main Claim on Canonical Form Assignment Project Exam Help

- Theorem 1 If ~~message size~~ ~~tations~~ are not bounded, every deterministic  $r$ -round algorithm can be transformed into an algorithm having the canonical form (i.e., it is possible to first communicate for  $r$  rounds and then do all the computations in the end). **Add WeChat edu\_assist\_pro**
- Notice the import <https://eduassistpro.github.io/> ges of arbitrary size:
  - this size will depend on the number of rounds, and
  - it can be exponential in  $r$
- To handle “large size messages” you need “large memory”

A message in a distributed system can be anything I want to be!  
 Provided that I do not exceed memory requirements!

<https://eduassistpro.github.io/>

## Main Argument Assignment Project Exam Help

- Consider an  $r$ -round algorithm  $\mathcal{A}$  and want to show that  $\mathcal{A}$  can be brought to a canonical form.  
[Add WeChat edu\\_assist\\_pro](#)
- First, let the nodes communicate for  $r$  rounds.
- Assume that in every round, every node sends its complete state to all of its neighbors.  
[Assignment Project Exam Help](#)
- By induction, after  $r$  rounds, the initial state of all other nodes at distance at most  $r$  from  $v$  will be known to  $v$ .  
[Add WeChat edu\\_assist\\_pro](#)
- Hence, after  $r$  rounds, a node  $v$  will have had initial knowledge of all the nodes in its  $r$ -neighborhood.
- We can show that this suffices to simulate locally (at node  $v$ ) enough of Algorithm  $\mathcal{A}$  to compute all the messages that  $v$  receives in the  $r$  communication rounds of a regular execution of Algorithm  $\mathcal{A}$ .

<https://eduassistpro.github.io/>

## Assignment Project Exam Help

- It is straightforward to generalize to randomized algorithms
  - Every node first computes all the random bits it needs throughout the algorithm.
- The random bits are then part of the initial state of a node.

In Canonical form of distributed computation we aggregate the messages into trees.

<https://eduassistpro.github.io/>

Assignment Project Exam Help

Add WeChat edu\_assist\_pro

Assignment Project Exam Help

**Dis** <https://eduassistpro.github.io/> **WS**

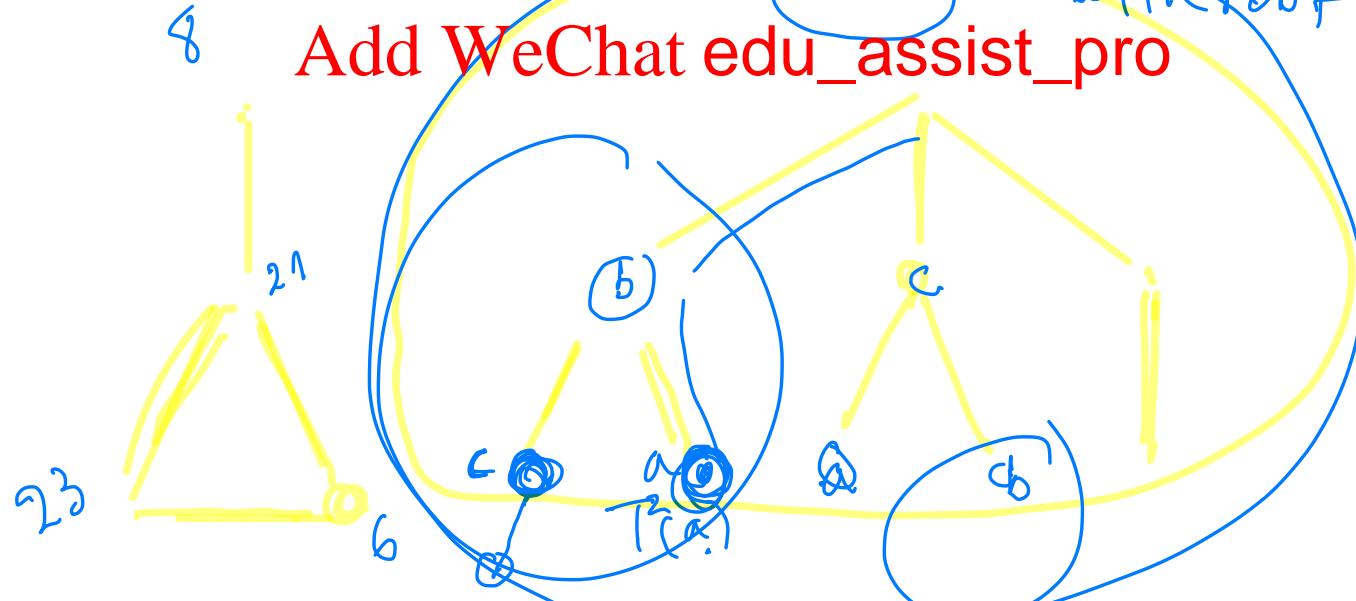
Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

Views: Undirected Networks  
**Assignment Project Exam Help**

- Each node has port labels and can simulate its knowledge. [Add WeChat edu\\_assist\\_pro](#)
- The view of depth  $k$  of a node is a tree containing information on all the walks of length  $k$  leaving that node.
- Views contain all the information that nodes could obtain by exchanging messages. [Assignment Project Exam Help](#)

<https://eduassistpro.github.io/> of depth 2 with root a



<https://eduassistpro.github.io/>

## $r$ -View (or $r$ -Hop View or $r$ -Neighborhood) Assignment Project Exam Help

- Collection of initial states of all  $n$   $r$ -neighborhood of a node  $v$ , is called  $r$ -hop  $\nabla$  neighborhood) of  $v$ .
  - For a given graph  $G$ , it is denoted by

Assignment Project Exam Help

$V_r^G(v)$  or  $N_r^G(v)$

- We usually omit  $G$  (ar from the context) and denote it by

Add WeChat edu\_assist\_pro

$V_r(v)$  or

- A view can be enriched as needed by including information:
  - on node states,
  - node topology  $r$  hops away from the source  $v$ ,
  - etc

satan

<https://eduassistpro.github.io/>

Example:  $r$ -Hop Views for  $r = 0, 1, 2, 3$   
Assignment Project Exam Help

Add WeChat edu\_assist\_pro

2-Hop

Assignment Project Exam Help

1-Hop

<https://eduassistpro.github.io/>

0-Hop

Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

## Assignment Project Exam Help

- Assume that initially, every node has its label (identifier) and potentially some edge, its label (identifier) and potentially so put.
- The  $r$ -hop view of a node  $v$  then includes
  - the complete topology of the  $r$ -neighborhood,
  - possibly edge  $e$  in the subgraph, and <https://eduassistpro.github.io/>
  - the labels and additional inputs of all nodes in the  $r$ -neighborhood.

<https://eduassistpro.github.io/>

## The View as a Function Assignment Project Exam Help

- **Theorem 2** A deterministic algorithm  $A$  is a function that maps every possible set of possible outputs.
- By Theorem 1, we know that we can transform Algorithm  $A$  to the canonical form.  
**Assignment Project Exam Help**
- After  $r$  communication,  $v$  knows exactly its  $r$ -hop view.  
**Add WeChat edu\_assist\_pro**
- This information suffices to compute the view  $v$ .

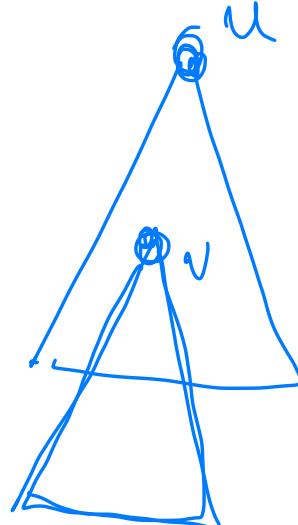
<https://eduassistpro.github.io/>

## Assignment Project Exam Help

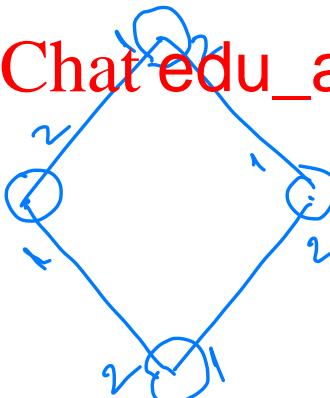
### Issues

- Two nodes with equal  $r$ -  
– have to compute the same  $r$ -round  
algorithm.
- For coloring algorithms, the only input of a node  $v$  is its label.  
– The  $r$ -hop  $v_1$   
 $r$ -neighbor

## Assignment Project Exam Help



Add WeChat edu\_assist\_pro



propal  
verhces  
will have  
identical  
views

<https://eduassistpro.github.io/>

## Assignment Project Exam Help

Power of Views<sup>a</sup>

A graph of  $n$  nodes

- For a graph of  $n$  nodes,  $N$  have the same view of depth  $r$  if they have the same views for all depths.

Aside  
not required

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

- Taking the diameter  $\delta$  of a graph into account, can improve  $n - 1$  to

$$O(\delta + \delta \log(n/\delta))$$

for bidirectional graphs with port numberings

---

<sup>a</sup>We won't discuss details for these claims.

<https://eduassistpro.github.io/>

## Views: Directed Networks Assignment Project Exam Help

- A view can be computed by a node o  
distributed deterministic al

Add WeChat edu\_assist\_pro

Is there a  
theorem on  
the power of  
for directed  
Communication  
 $\log(n)$ .

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

Directed graph  
build their  
views in a  
chronal  
manner

- In directed networks we have “in” and “out” views at a node.

<https://eduassistpro.github.io/>

Assignment Project Exam Help  
Add WeChat ~~edu\_assist\_pro~~

two hops

Assignment Project Exam Help

<https://eduassistpro.github.io/g>

Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

## Broadcasting Assignment Project Exam Help

- Broadcasting refers to a method of sending a message to all recipients ‘at once’ in a network.

## Assignment Project Exam Help

<https://eduassistpro.github.io/>

[Add WeChat edu\\_assist\\_pro](#)

- It is initiated by a single processor, the source.
- The source sends a message to all other nodes in the system.
- In a typical network it may not be possible to send a message “at once” since there might be multiple hops from the source to the rest of the nodes.

<https://eduassistpro.github.io/>

## Graph Concepts in Broadcasting Assignment Project Exam Help

- The distance  $d(u, v)$  between nodes  $u$  and  $v$  in an undirected graph  $G$  is the number of hops in the shortest path between  $u$  and  $v$ .

- The radius

- of a node  $u$  is the maximum distance between  $u$  and any other node in the graph.

<https://eduassistpro.github.io/>

Add WeChat [edu\\_assist\\_pro](https://eduassistpro.github.io/)

- of a graph is the minimum radius of any node in the graph.

$$R = \min_u R(u)$$

- The radius and diameter of a graph are called graph eccentricities.



A hand-drawn directed graph on grid paper with four nodes. Node 1 (top-left) has a self-loop and an edge to node 2. Node 2 (top-center) has an edge to node 3 (bottom-center). Node 3 has an edge to node 4 (top-right), a self-loop, and an edge back to node 2. Node 4 has a self-loop and an edge back to node 3.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

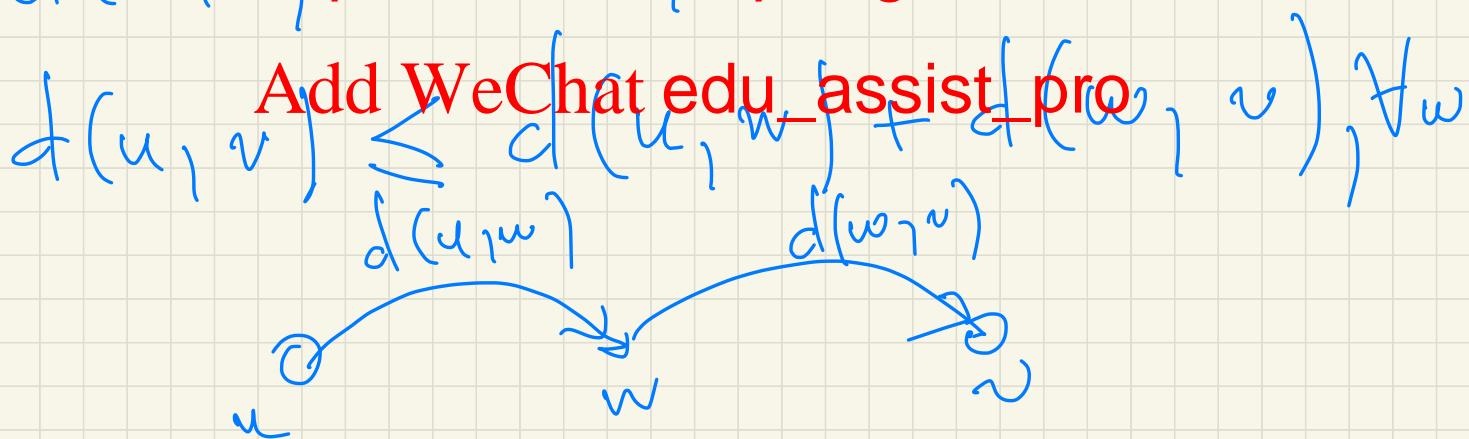
Add WeChat edu\_assist\_pro

$d(u, v)$

$d_G(u, v)$  = ~~# of hops from~~ <sup>min</sup> ~~u to v~~

$d(u, v)$  ~~Assignment Project Exam Help~~

$d(u, v) \leq d(u, w) + d(w, v)$   
<https://eduassistpro.github.io/>

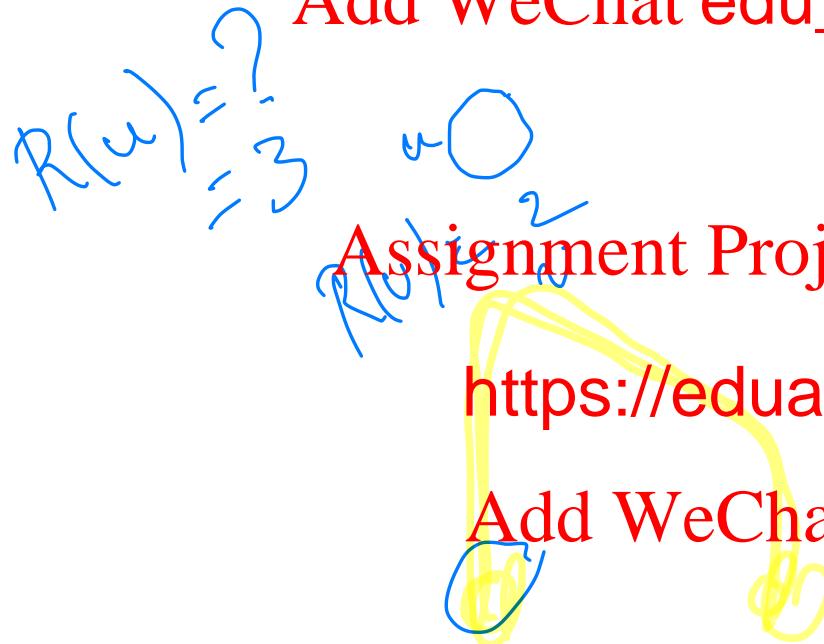


<https://eduassistpro.github.io/>

Examples: Graph Eccentricities  
Assignment Project Exam Help

- Distance

Add WeChat edu\_assist\_pro



<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

- Radius
- Diameter

<https://eduassistpro.github.io/>

Examples: Graph Eccentricities  
Assignment Project Exam Help

- Radius, Diameter

Add WeChat edu\_assist<sub>pro</sub>  $R(u)$



<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

- There is a close relationship between the radius  $R$  and the diameter  $D$  of a graph

$$- R \leq D \leq 2R.$$

$$R \leq D \leq 2R$$

<https://eduassistpro.github.io/>

Examples: Graph Eccentricities  
**Assignment Project Exam Help**

- What are the Radius and Diameter?

Add WeChat [edu\\_assist\\_pro](#)

**Assignment Project Exam Help**

<https://eduassistpro.github.io/>

Add WeChat [edu\\_assist\\_pro](#)

<https://eduassistpro.github.io/>

Assignment Project Exam Help

Add WeChat edu\_assist\_pro

Assignment Project Exam Help

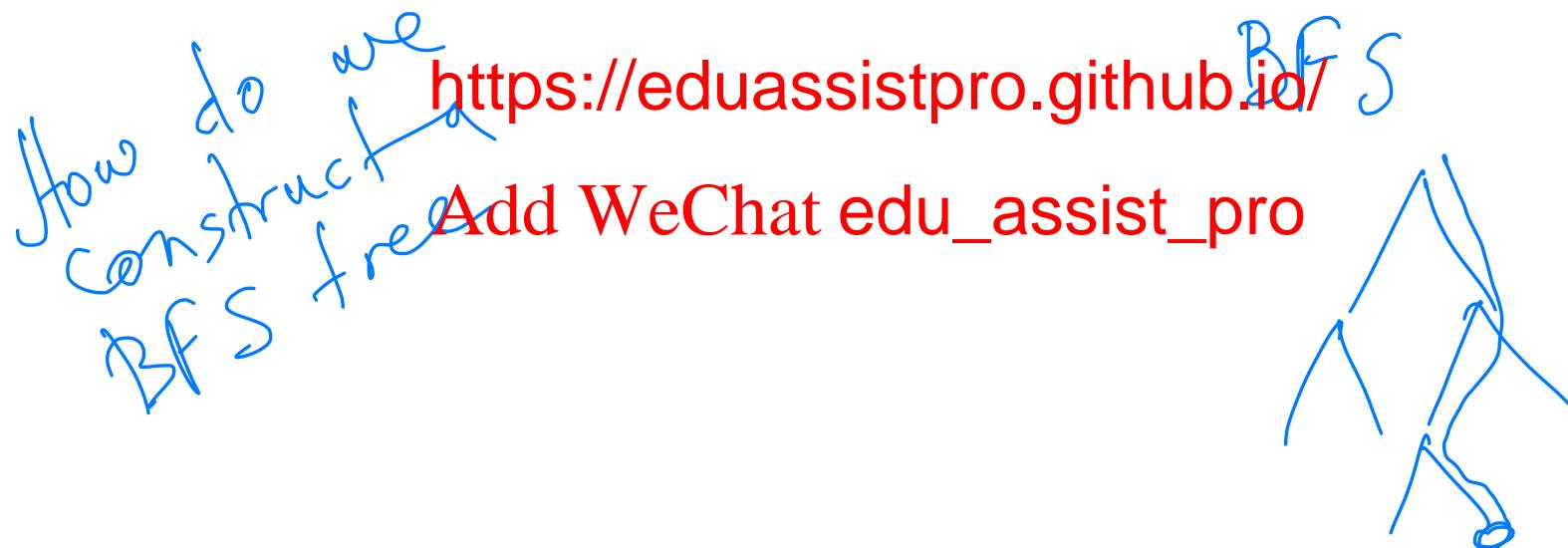
<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

## BFS Spanning Trees Assignment Project Exam Help

- Traversal of a graph is performed in some predefined order.
- **Breadth-First-Search Tree.** A breadth-first-search tree  $T$  of a graph  $G$  is a spanning tree of  $G$  such that for every node of  $G$ , the tree path is a minimum-hop path to the root.



- Of course a root must be specified!

<https://eduassistpro.github.io/>

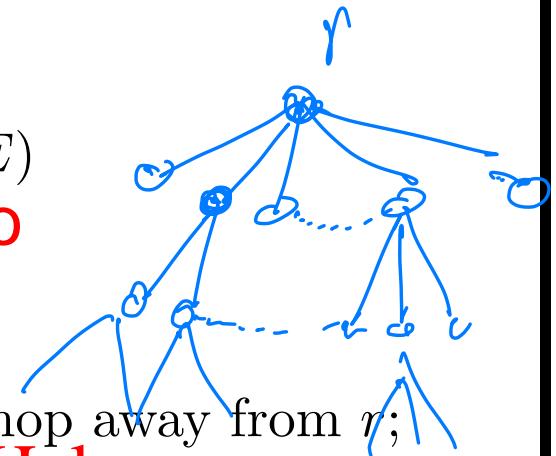
## BFS Algorithm<sup>a</sup> Assignment Project Exam Help

- BFS Algorithm: Input  $(V, E)$

Proceed by layers,  
[Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/)

1. mark the root  $r$ ;
2. mark all neighbor vertices that are one hop away from  $r$ ;
3. mark new vertices that are one hop away from these neighbors (the  $r$ );  
<https://eduassistpro.github.io/>
4. and so on.

- It uses a FIFO queue
- It checks whether a vertex has been discovered before enqueueing the vertex rather than delaying this check until the vertex is dequeued from the queue




---

<sup>a</sup>Invented in 1945 by Konrad Zuse

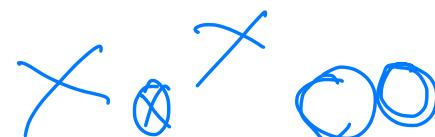
<https://eduassistpro.github.io/>

## BFS Algorithm Assignment Project Exam Help

- How do you construct a BFS tree f

?

Add WeChat [edu\\_assist\\_pro](https://eduassistpro.github.io/)

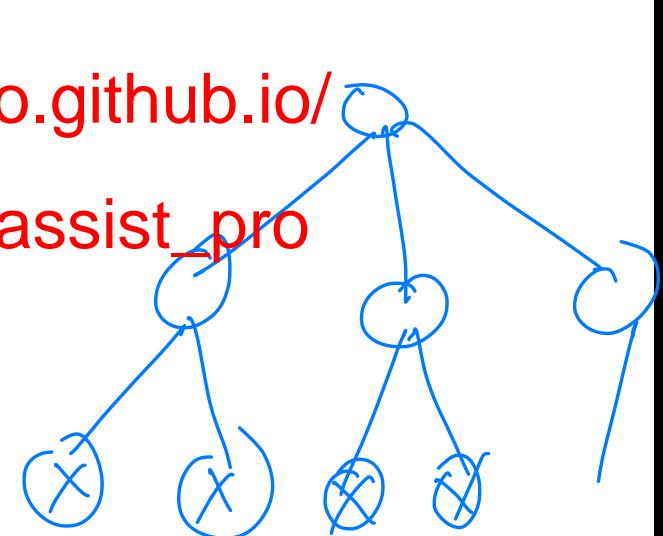


[Assignment Project Exam Help](https://eduassistpro.github.io/)

<https://eduassistpro.github.io/>

- First choose any vertex as a root!

Add WeChat [edu\\_assist\\_pro](https://eduassistpro.github.io/)



<https://eduassistpro.github.io/>

## BFS (Distance Computation (1/2)) Assignment Project Exam Help

- It starts by placing the source node  $s$  at distance  $d(s) = 0$ ; the distance of all other nodes starts  $d(i) = \infty$ .
- At the  $k$ th step (starting at  $k = 0$ ), all nodes  $i$  at distance  $d(i) = k$  are examined, and any neighbors  $j$  with  $d(j) = \infty$  (i.e., not yet discovered) have their distance  $d(j)$  set to  $k + 1$ .
- The process halts when all nodes have been examined;  $d(j)$  is then the length of the shortest path from  $s$  to  $j$ , or  $d(j) = \infty$  if there is no such path.

if no path exist  
 it belongs to a  
 different connected comp-

<https://eduassistpro.github.io/>

## BFS (Distance Computation (2/2)) Assignment Project Exam Help

- BFS is the simplest way to search
- It is suited only for unweighted graphs
- Example 1:

Assignment Project Exam Help

<https://eduassistpro.github.io/>

- Example 2: In a social network, you are at level one and your friends of friends are at level two in a BFS starting at your node.

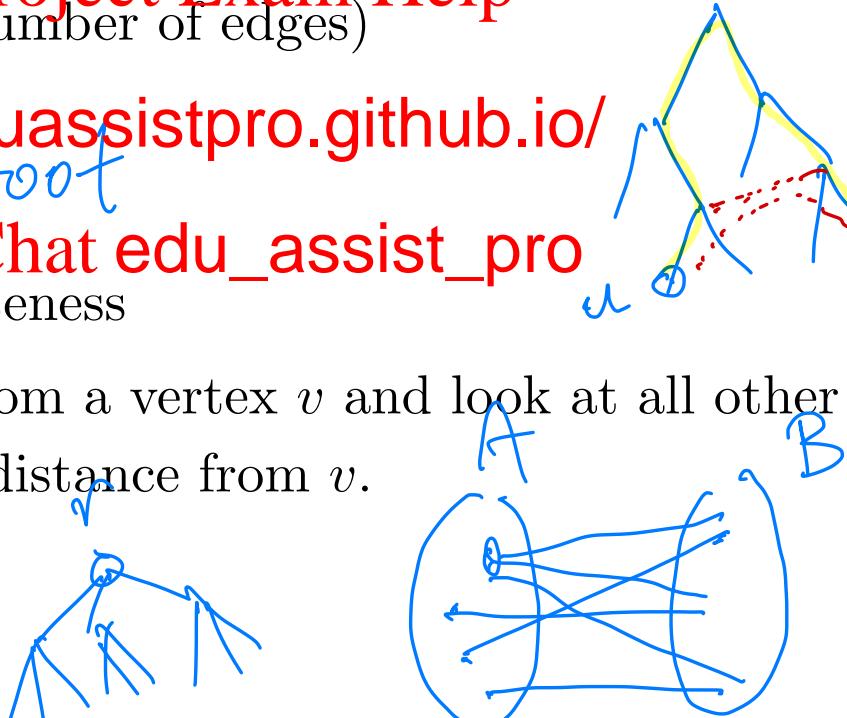
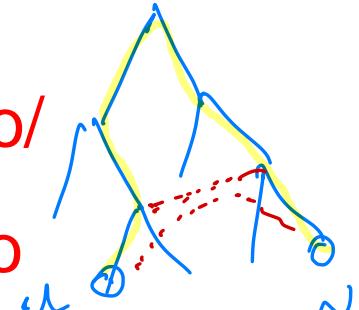
<https://eduassistpro.github.io/>

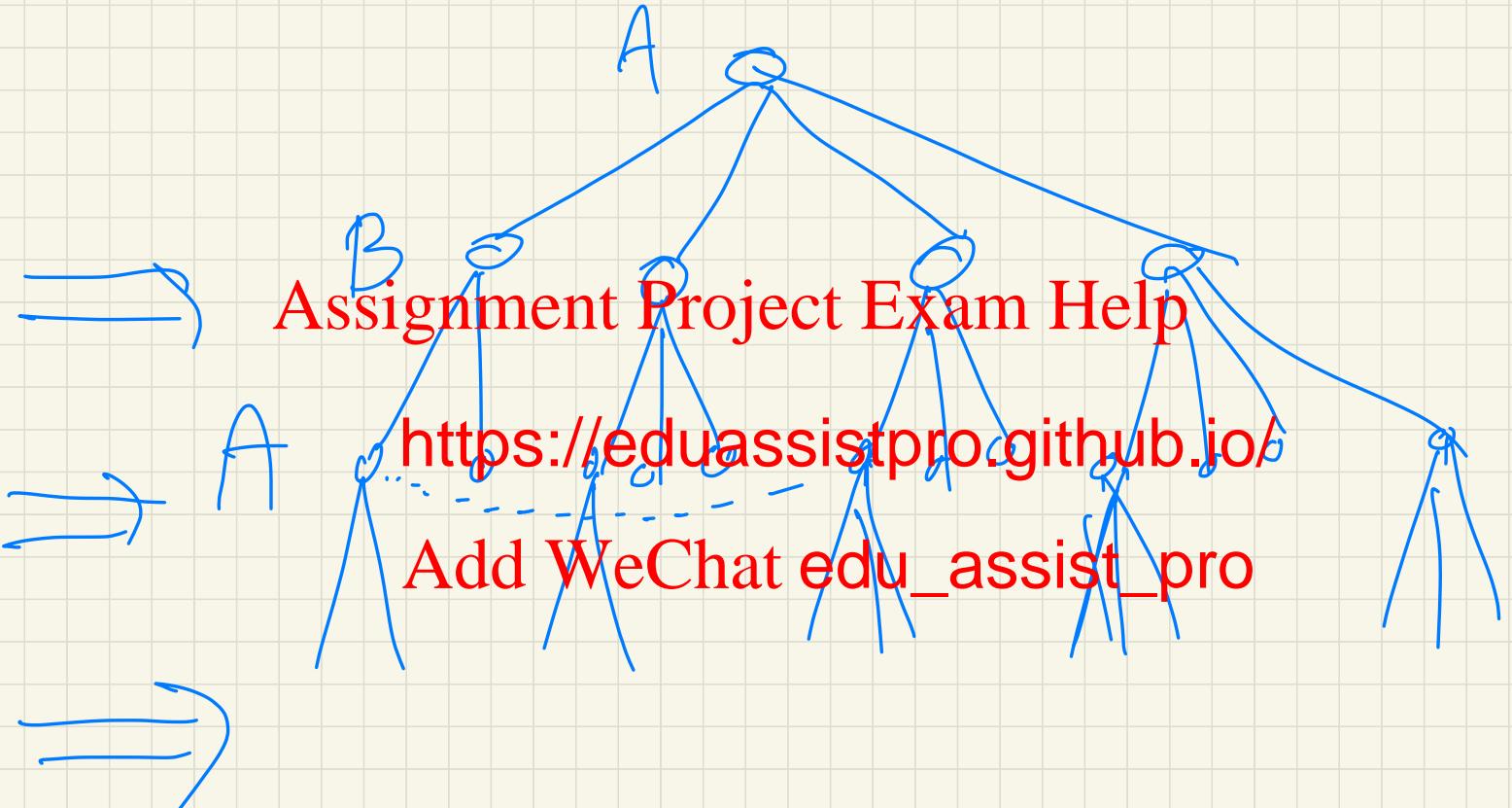
What is BFS Tree Used for?

## Assignment Project Exam Help

- Finding all nodes within one component
  - BFS by itself is not enough: so sing is needed!
- Finding the shortest path between two nodes  $u$  and  $v$  (with path length measured by number of edges)
  - $u$  and  $v$  could be respectively.
- Testing a graph for bipartiteness
  - Construct a BFS tree from a vertex  $v$  and look at all other vertices at odd or even distance from  $v$ .
- Doing efficient broadcast
  - from any node.

Add WeChat edu\_assist\_pro  
forget it ??





<https://eduassistpro.github.io/>

## DFS Spanning Trees Assignment Project Exam Help

- For a rooted spanning tree

Add WeChat [edu\\_assist\\_pro](https://eduassistpro.github.io/)

Assignment Project Exam Help

<https://eduassistpro.github.io/>

let us denote by Add WeChat [edu\\_assist\\_pro](https://eduassistpro.github.io/)

- $S(u)$  all the nodes in the subtree of  $u$ , and
- $P(u)$  denote all the vertices that exist in a path between  $u$  and the root.

<https://eduassistpro.github.io/>

## DFS Spanning Trees Assignment Project Exam Help

- DFS Algorithm: Input  $(V, E)$

1. Start from a vertex  $r$ ;

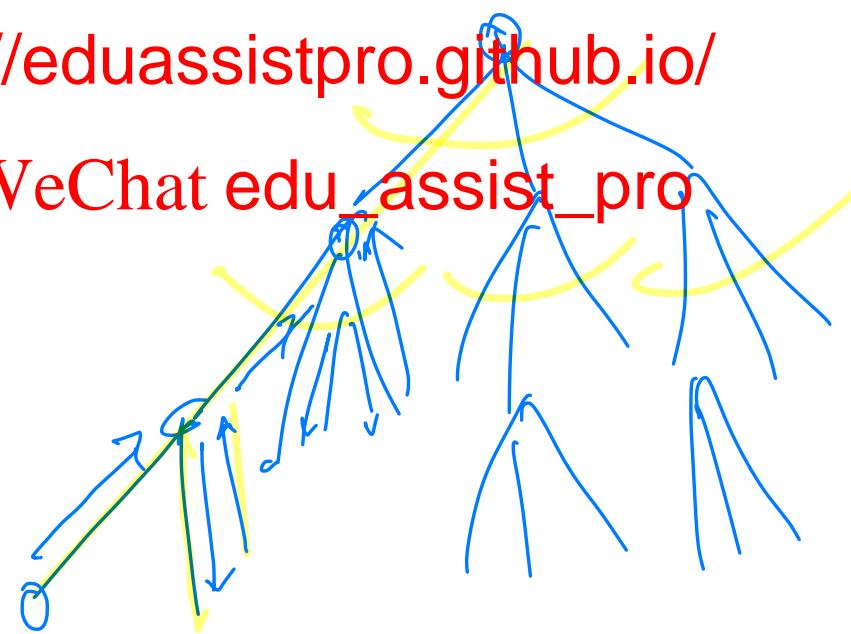
2. visit all possible vertices as far as you can reach;

3. when all vertices are visited, return to the current parent node.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro



<https://eduassistpro.github.io/>

## DFS (Depth-First Search) (1/2) Assignment Project Exam Help

- DFS visits the same nodes as BFS in a different order.
- If it sees an unvisited node  $j$  while examining node  $i$ , it fully discovers all unvisited nodes reachable from  $j$  and then backtracks to node  $i$  to consider the remainder of the nodes adjacent to  $i$ .

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

- It is best described recursively.

<https://eduassistpro.github.io/>

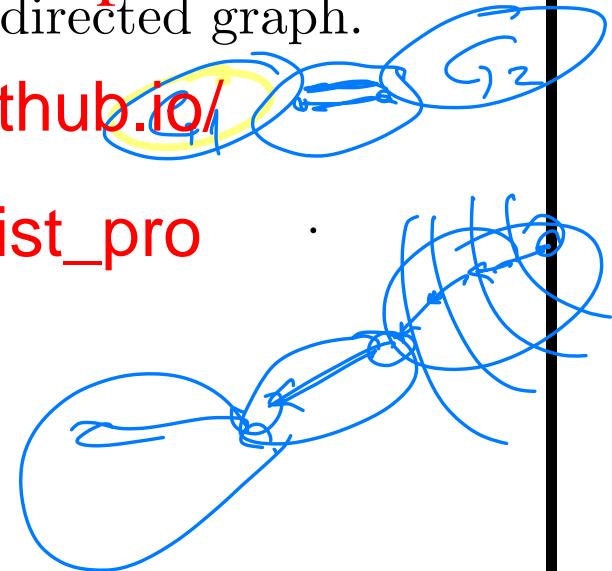
## DFS (Depth-First Search) (2/2) Assignment Project Exam Help

- All nodes start out unvisited.
- DFS( $i$ ):
  1. mark  $i$  as visited
  2. for all nodes  $j$  adjacent to  $i$  do:  
<https://eduassistpro.github.io/>  
Assignment Project Exam Help
  3. if node  $j$  is not visited  $DFS(j)$
- Example      <https://eduassistpro.github.io/>  
<https://eduassistpro.github.io/>  
Assignment Project Exam Help

<https://eduassistpro.github.io/>

## What is DFS Tree Used for? Assignment Project Exam Help

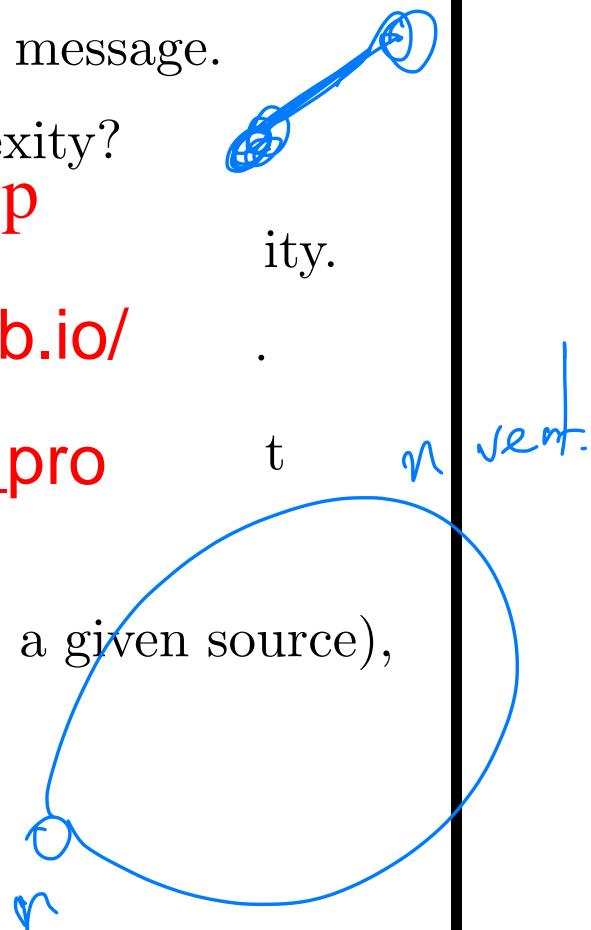
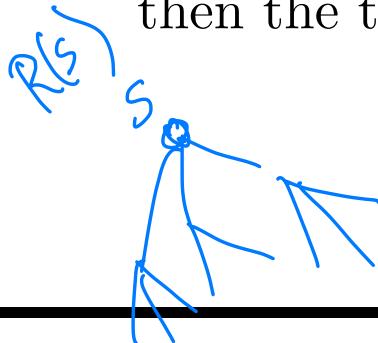
- Finds all of the vertices reachable from vertex  $r$  in a graph [Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/)
  - unlike BFS it does not need to search the whole graph.
- Topological sorting. [Assignment Project Exam Help](https://eduassistpro.github.io/)
  - this is because of the way it traverses a directed graph.
- Finding the bridges <https://eduassistpro.github.io/>
  - these are edges whose removal disconnects the graph. [Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/)
- Finding connected components.
  - like BFS.
- Finding strongly connected components.
  - these are maximal “strongly connected components” of a directed graph.



<https://eduassistpro.github.io/>

## Lower Bounds Assignment Project Exam Help

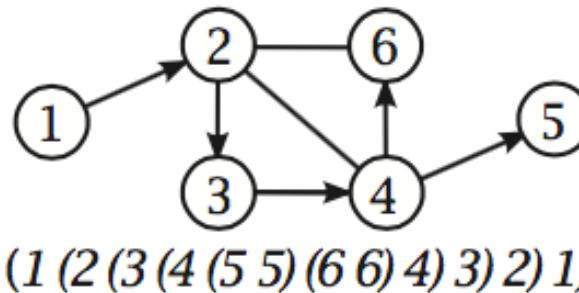
- The message complexity of broadcast in a  $n$  node graph is at least  $n - 1$ . [Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/)
  - This is because every node must receive the message.
  - Which graphs require  $n - 1$  message complexity?
- The source's radius is .
- You can use a pre-computed spanning tree with tight message complexity. [Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/)
- If the spanning tree is a BFS spanning tree (for a given source), then the time complexity is tight as well.



<https://eduassistpro.github.io/>

## More on BFS and DFS Assignment Project Exam Help

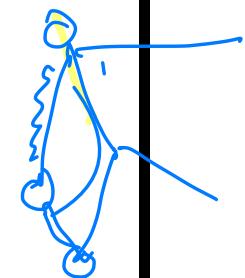
- Both BFS and DFS describe a tree.  $i$  is the parent of  $j$  if the unvisited node  $j$  is discovered from node  $i$ .
- The DFS tree has a rich set of mathematical properties.
  - For example, if “ $(i)$ ” is printed at the start of  $DFS(i)$  and “ $i)$ ” when it finishes (after traversing all its neighbors  $j$ ), then the result is a matching pair.
  - The parentheses of two nodes either nested one within the other, or they are disjoint.



<https://eduassistpro.github.io/>

## Impact of Knowledge: Clean Graphs Assignment Project Exam Help

- If the graph is stored in adjacency list form, then BFS and DFS take an amount of time proportional to the size of the graph:  $O(|V| + |E|)$ , where  $|V|$  and  $|E|$  are the number of nodes and edges, respectively.
- Knowledge can affect the message complexity!
- Call a graph (network) clean if all nodes know the topology of the graph.
- If the nodes do not know the topology of the graph (in a clean network) then the number of edges is a lower bound for the broadcast message complexity.
  - If you do not try every edge, you might miss a whole part of the graph behind it.



<https://eduassistpro.github.io/>

Assignment Project Exam Help

Add WeChat edu\_assist\_pro

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

## Assignment Project Exam Help

- Flooding
- FloodMaxID
- OptFloodMax

Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

## Flooding Assignment Project Exam Help

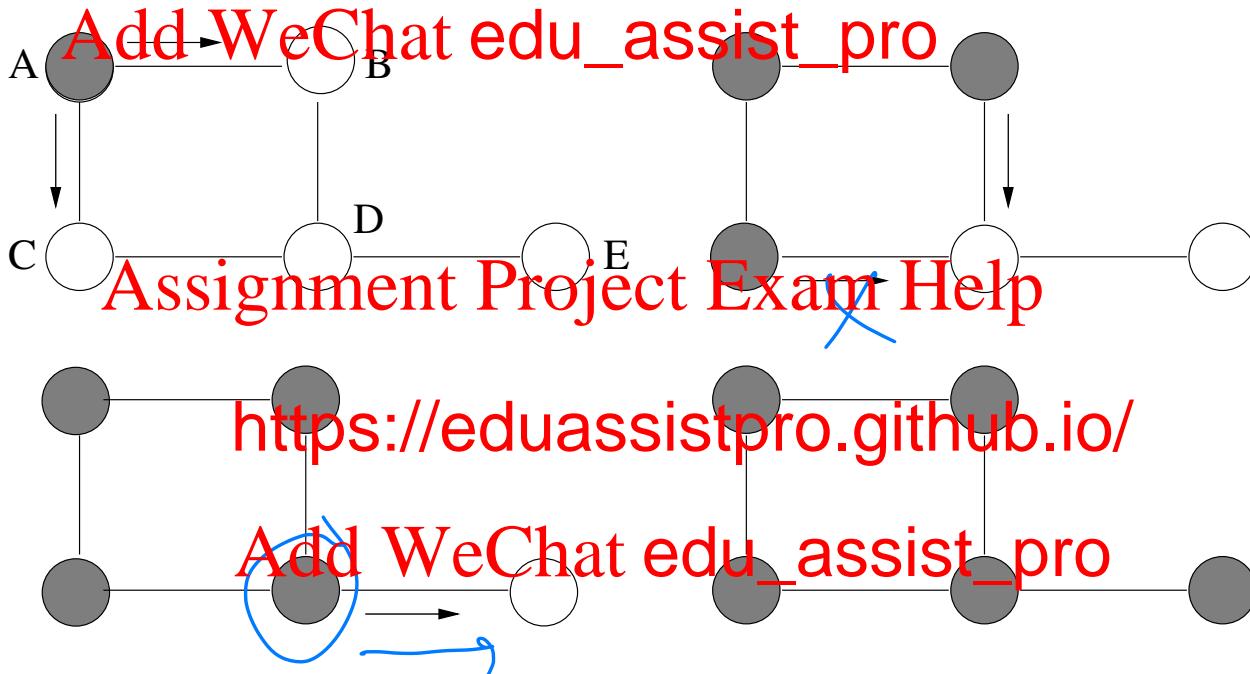
- Used by nodes to identify thems
- **Flooding Algorithm**
  1. The source (root) sends the message to all neighbors.
  2. Each other node  $v$  upon receiving the message the first time forwards the m
  3. Upon later rec <https://eduassistpro.github.io/> r edges), a node can discard the message.

<https://eduassistpro.github.io/>  
**Add WeChat edu\_assist\_pro**

<https://eduassistpro.github.io/>

## Flooding Example Assignment Project Exam Help

- Let  $A$  be the initiating node:



- Note that node  $D$  receives two messages.

<https://eduassistpro.github.io/>

## Flooding and Trees Assignment Project Exam Help

- If node  $v$  receives the message  $u$ , then node  $v$  calls node  $u$  parent.
  - Parent relation defines a spanning tree  $T$  (nodes receiving more than one message keep message only from one initiator).
  - If flooding algorithm terminates, then  $T$  is a BFS spanning tree.
- Let  $R(s)$  be the radius of the source network.
  - In asynchronous systems the flooding algorithm terminates after  $R(s)$  time units.
  - However, the spanning tree constructed may not be a BFS spanning tree.

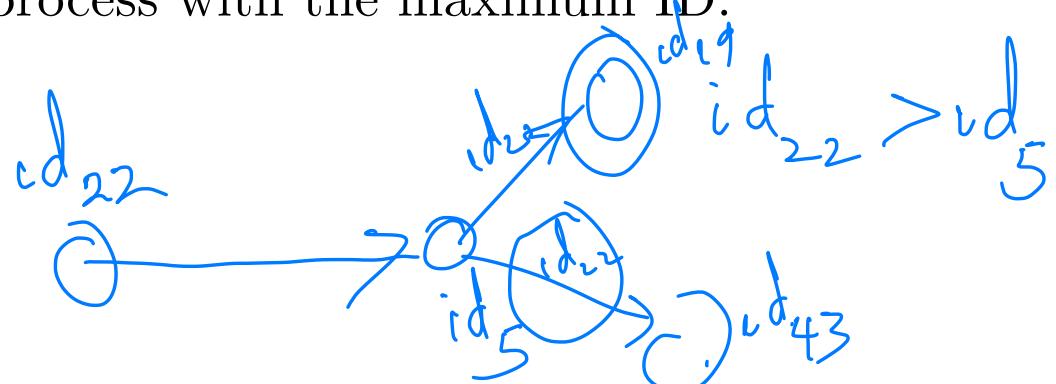
<https://eduassistpro.github.io/>

## FloodMaxID Assignment Project Exam Help

- We give a simple algorithm that  
non-leaders to identify them  
[Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/)
- The algorithm
  - requires that the processes know the diameter of the network;
  - floods the max  
[Assignment Project Exam Help](https://eduassistpro.github.io/)
    - \* so we call it the FloodMaxID algori  
[Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/)
- The algorithm makes leader election po  
network.

<https://eduassistpro.github.io/>

- Flood MaxID
- Assignment Project Exam Help
- FloodMaxID Algorithm
  1. Every process maintains a variable  $\text{MaxID}$  which is the maximum ID it has seen so far (initially its own).
  2. At each round, each process propagates this maximum on all of its outgoing edges.
  3. After  $D$  (di) steps, the process with the maximum ID is the leader; otherwise, it is a non-leader.
- FloodMax elects the process with the maximum ID.



<https://eduassistpro.github.io/>

## Analysis of FloodMax Assignment Project Exam Help

- Define  $i_{\max}$  to be the index of the maximum ID, and [Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/)
- **Theorem 3** In the FloodMax algorithm, process  $i_{\max}$  outputs leader and each other process outputs non-leader, within diameter rounds.
- **Main Claim** <https://eduassistpro.github.io/>
  - $status_{i_{\max}} = \text{leader}$  and [Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/)
  - $status_j = \text{non-leader}$ , for every  $j \neq i_{\max}$ .
- The key to the proof of this Claim is the fact that
  - after  $r$  rounds, the maximum ID has reached every process that is within distance  $r$  of  $i_{\max}$ , as measured along directed paths in  $G$ .

in diameter numbered  
 of steps who the leader if  $i_{\max}$

Assignment Project Exam Help

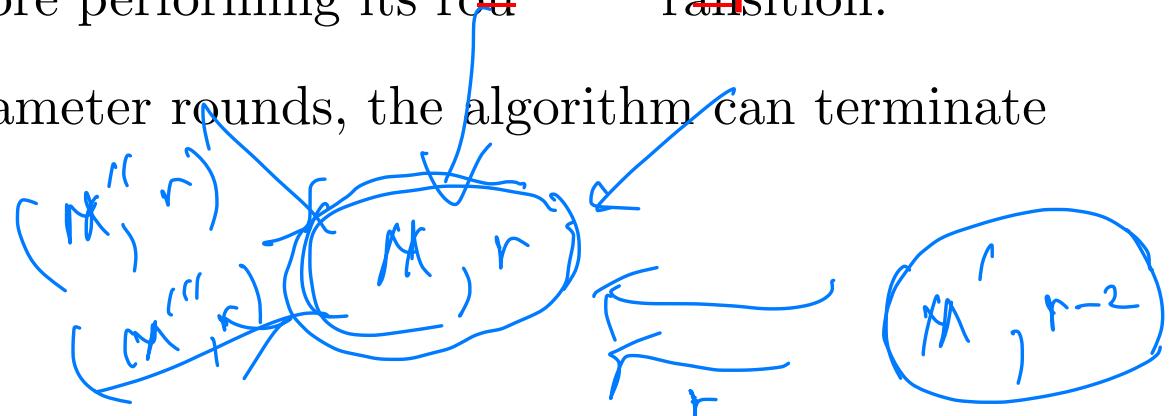
<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

## FloodMax Assignment Project Exam Help

- The FloodMax algorithm does not try to simulate the rounds asynchronously, because it ends in the asynchronous model.
- However, it is possible to simulate the rounds asynchronously.
  - We simply require each process that sends a round  $r$  message to tag it with  $r$ .
  - The recipient waits to receive round messages from all its neighbors before performing its round transition.
- By simulating diameter rounds, the algorithm can terminate correctly.



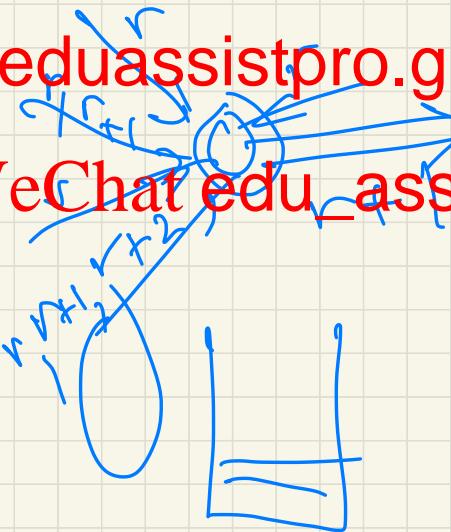
In a way, we make the algorithm "message driven"

Assignment Project Exam Help

I must messages of round  $r$  from all my neighbors before I proceed to  $rt$ .

<https://eduassistpro.github.io/>

Add WeChat `edu_assist_pro`



<https://eduassistpro.github.io/>

## OptFloodMax Algorithm Assignment Project Exam Help

- There is a simple improvement to decrease the communication complexity, although it does not decrease the order of magnitude in the worst case.
- Namely, processes can send their current max user ID values only when they first learn about them, not at every round.

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

Assignment Project Exam Help

Add WeChat edu\_assist\_pro

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

## Convergecast Assignment Project Exam Help

- Convergecast is reversed bro
  - Instead of a root sending a message, all other nodes send information to a root.

## Assignment Project Exam Help

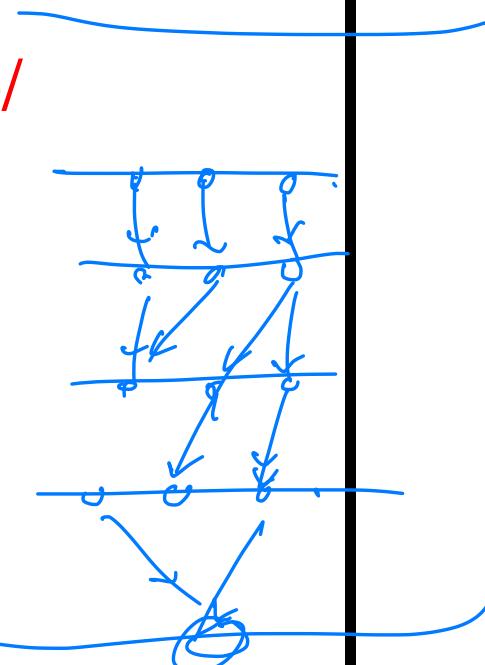
<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

- Convergecast is useful for input collection.

garbage

- 5C



<https://eduassistpro.github.io/>

## Echo Algorithm Assignment Project Exam Help Echo Algorithm

- **Requirement:** Add WeChat `edu_assist_pro` at the leaves.
  1. A leaf sends a message to its parent.
  2. If an inner node has received a message from each child, it sends a message to the parent.



<https://eduassistpro.github.io/>

## Complexity Issues: Broadcast and Convergecast (1/2)

### Assignment Project Exam Help

- The echo algorithm is paired with the convergecast algorithm, which is used to let the leaves know when to start the echo process; this is known as flooding/echo.
- One can use convergecast for termination detection.
  - If a root wants to know whether all nodes in the system have finished some task:
    - \* the message in the echo algorithm tells whether the subtree has finished the task.
- Message complexity of the echo algorithm is  $n - 1$ ,
  - but together with flooding it is  $O(m)$ , where  $m = |E|$  is the number of edges in the graph.

<https://eduassistpro.github.io/>

## Complexity Issues: Broadcast and Convergecast (2/2)

### Assignment Project Exam Help

- The time complexity of the echo is determined by the depth of the spanning tree (i.e., the maximum distance from a node to the root within the tree) generated by the flooding algorithm.
- The flooding/echo algorithm can do much more than collecting acknowledgements from subtrees.
  - For instance, it can compute the sum of values stored in the system, or the maximum ID (for all nodes), or the sum of all values stored in the system.
- By combining results one can compute even fancier aggregations, e.g., with the number of nodes and the sum one can compute the average. With the average one can compute the standard deviation. And so on . . .

<https://eduassistpro.github.io/>

# Application to Leader Election Assignment Project Exam Help

- Asynchronous broadcast algorithms can be used to solve the leader election problem in an ad-hoc network.
    - without any distinguished source node and
    - without the processes having any knowledge of the number of nodes or the diameter of the network.
  - The processes need to know the number of nodes in the network.
    - Add WeChat to edu\_assist\_pro
    - Assignment Project Exam Help

Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

## Basic Leader Election Algorithm Assignment Project Exam Help

- Every node can initiate
  - first a broadcast, and
  - next a convergecast

in order to discover the maximum user ID in the network.

## Assignment Project Exam Help

- The node that finds  $t$  elects itself as leader
- This algorithm uses  $O(n|E|)$  messages

<https://eduassistpro.github.io/>

Assignment Project Exam Help

Add WeChat `edu_assist_pro`

(Dir [Assignment Project Exam Help](https://eduassistpro.github.io/) FS

<https://eduassistpro.github.io/>

and [Applic](#)

Add WeChat `edu_assist_pro`

<https://eduassistpro.github.io/>

## Construction of BFS Assignment Project Exam Help

- How do we perform breadth-first search in a network based on an arbitrary strongly connected graph having a distinguished source node?
- We consider how to establish a breadth-first spanning tree for the (di)-graph.
- Motivation for <https://eduassistpro.github.io/> e desire to have a convenient structure to use as a base for communication.
- The BFS tree minimizes the maximum communication time from the process at the distinguished node to all other processes in the network
  - To do this run BFS from each node of the graph and compare values obtained at each node.

<https://eduassistpro.github.io/>

## Construction of BFS Assignment Project Exam Help

- We suppose that the network is static and that there is a distinguished source node  $i_0$ .  
[Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/)
- The algorithm is supposed to output the structure of a breadth-first directed spanning tree of the network graph, rooted at  $i_0$ .  
[Assignment Project Exam Help](https://eduassistpro.github.io/)
- The output should be a directed spanning tree where every process other than  $i_0$  should have a parent node that gets set to indicate the node that is its parent.  
[Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/)
- As usual, processes only communicate over directed edges.
- Processes are assumed to have user IDs but to have no knowledge of the size or diameter of the network.

<https://eduassistpro.github.io/>

## SynchBFS Assignment Project Exam Help

- The basic idea for this algorithm is standard sequential breadth-first search algorithm.
- **SynchBFS Algorithm**
  1. At any point during execution, there is some set of processes that is *marked*: initially just  $i_0$ .
  2. Process  $i_0$  sends a search message to all of its outgoing neighbors.
  3. At any round, if an unmarked process receives a search message, it marks itself and chooses one of the processes from which the search has arrived as its parent.
  4. At the first round after a process gets marked, it sends a search message to all of its outgoing neighbors.

<https://eduassistpro.github.io/>

## Analysis of SynchBFS Assignment Project Exam Help

- We can prove the invariant that
  - after  $r$  rounds, every node  $i_0$  from  $i_0$  in the graph,  $1 \leq d \leq r$ , has its parent pointer defined; moreover, each such pointer points to a node at distance  $d - 1$  from  $i_0$ .

This invariant can, as usual, be proved by induction on the number of rounds

<https://eduassistpro.github.io/>

- The time complexity is at most diameter  $r$
- The number of messages is just
  - a search message is transmitted exactly once on each directed edge.

<https://eduassistpro.github.io/>

## Applications of BFS Assignment Project Exam Help

- Message Broadcast
- Global computation
- Electing a leader
- Computing the diameter

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

<https://eduassistpro.github.io/>

## Message Broadcast: Piggybacking Assignment Project Exam Help

- The SynchBFS algorithm can be modified to implement message broadcast.  
<https://eduassistpro.github.io/>
- If a process has a message  $M$  that it wants to communicate to all of the processes in the network,
  - it can simply initiate an execution of SynchBFS with itself as the root, piggybacking <https://eduassistpro.github.io/> in the search message it sends in round-trip.
- Other processes continue to piggyback <https://eduassistpro.github.io/> in their search messages as well.
  - Since the tree eventually spans all the nodes, message  $M$  is eventually delivered to all the processes.

<https://eduassistpro.github.io/>

## Global Computation Assignment Project Exam Help

- This means
  - Collection of information from the network or,
  - more generally, the computation of a function based on distributed inputs.

## Assignment Project Exam Help

- For example,
  - consider the problem of finding the sum of all the nonnegative integer input values in a network. Using a BFS tree, this can be done easily (and efficiently) as follows.

<https://eduassistpro.github.io/>

## Global Computation Assignment Project Exam Help

- Starting from the leaves, “fan in” procedure, as follows:  
[Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/)
  1. Each leaf sends its value to its parent;
  2. each parent waits until it gets the values from all its children, adds them to its own input value, and then sends the sum to its own parent.
- The sum calculated by the root of the BFS tree is the answer.  
[Add WeChat edu\\_assist\\_pro](https://eduassistpro.github.io/)

<https://eduassistpro.github.io/>

## Electing a Leader Assignment Project Exam Help

- Using SynchBFS, an algorithm to elect a leader in a network with IDs, even though processes have no knowledge of  $n$  or diameter.
  1. Namely, all the processes can initiate breadth-first searches in parallel.
  2. Each process determines and the global computation determines the maximum ID of any process in the network.
  3. The process with the maximum ID then declares itself to be the leader, and all others announce that they are not the leader.
- If the graph is undirected, the time is  $O(\text{diameter})$  and the number of messages is  $O(n|E|)$ .

<https://eduassistpro.github.io/>

## Computing the Diameter Assignment Project Exam Help

- The diameter of the network can be computed by having all processes initiate breadth-first search parallel.
  1. Each process  $i$  uses the tree thereby constructed to determine  $\max-dist$ , defined to be the maximum distance from  $i$  to any other process in the network.
  2. Each process computes  $\max-dist$  values.
- If the graph is undirected, the time is  $O(diam)$  and the number of messages is  $O(n|E|)$ , where  $diam$  is the diameter of the graph.
- The diameter thus computed could be used, for example, in the leader-election algorithm FloodMax.

<https://eduassistpro.github.io/>

## Exercises<sup>a</sup> Assignment Project Exam Help

1. Explain why every tree is a bipartite graph.
2. Let  $T$  be a graph with  $n$  vertices. The following statements are equivalent.
  - (a)  $T$  is connected and has no cycles.
  - (b)  $T$  has  $n-1$  edges and has no cycles.
  - (c)  $T$  is connected and has no cycles.
  - (d)  $T$  is connected and has  $n-1$  edges that connects  $T$ .
  - (e) Any two vertices of  $T$  are connected by one path.
  - (f)  $T$  contains no cycles, but the addition of one edge creates a cycle.
3. Give an algorithm to compute the diameter and radius of a tree.
4. Determine the size of a message which propagates for  $r$  hops in a complete graph  $K_n$ .

---

<sup>a</sup>Not to hand in!

<https://eduassistpro.github.io/>

## Assignment Project Exam Help

“Synchronous Algorithm C”. More specifically, consider a complete binary tree of height  $n$ . Label the ports at an interior node as  $L, R$  (for the Left and Right siblings at a node), and  $P$  for its parent. Do the same in an analogous manner for the root and the leaves. For each  $r \leq n$  and each node

## Assignment Project Exam Help

5. A connected graph includes every vertex exactly once (such a path is called Hamiltonian). A connected graph is semi-Hamiltonian if there is a path (but not a cycle) that includes every vertex exactly once (such a path is called semi-Hamiltonian). Determine which of the following graphs are semi-Hamiltonian, and write down a corresponding semi-Hamiltonian path where possible:

<https://eduassistpro.github.io/>

## Assignment Project Exam Help

Add WeChat [edu\\_assist\\_pro](https://edu_assist_pro)

6. A forest is a graph (not necessarily connected) each of whose components is a tree  
(a) Let  $G$  be a forest with  $k$  components. How many edges does  $G$  have?  
(b) Construct a forest with 12 vertices and  
(c) Is it true that every forest with  $k$  components has at least  $2k$  vertices of degree 1?
7. A spanning forest in a graph  $G$  (not necessarily connected) is obtained by constructing a spanning tree for each component of  $G$ .

<https://eduassistpro.github.io/>

- (a) Find a spanning forest for the following graph.

**Assignment Project Exam Help**

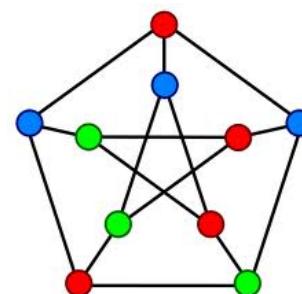
Add WeChat [edu\\_assist\\_pro](https://edu_assist_pro)

- (b) Let  $G$  be a graph, and let  $F$  be a subgraph of  $G$ . If  $F$  is a forest which includes all vertices of  $G$ , is  $F$  necessarily a spanning forest?

<https://eduassistpro.github.io/>

8. Find three spanning trees in the Petersen graph (shown below):

Add WeChat [edu\\_assist\\_pro](https://edu_assist_pro)



9. Prove that trees and forests are bipartite graphs.

<https://eduassistpro.github.io/>

10. Prove that, in a bipartite graph, every cycle has an even number of edges. Conversely, prove that, if every cycle of a graph has an even number of edges, then the graph is bipartite.

**Assignment Project Exam Help**  
**Add WeChat edu\_assist\_pro**  
Hint: Consider a connected graph  $G$ . Choose a vertex  $v$  in  $G$  and consider those vertices whose minimum distance from  $v$  is even and those whose minimum distance from  $v$  is odd. To which vertices are the “odd” vertices adjacent? To which vertices are the “even” vertices adjacent?

<https://eduassistpro.github.io/>

**Add WeChat edu\_assist\_pro**