THE UNIVERSITY OF MELBOURNE

COMP90038 Algorithms and Complexity

Assignment Project Exam Help

Lec versal

https://eduassistpro.github.io/

(with thanks to Hara ergaard)

Add WeChat edu_assist_pro

Toby Murray

toby.murray@unimelb.edu.au

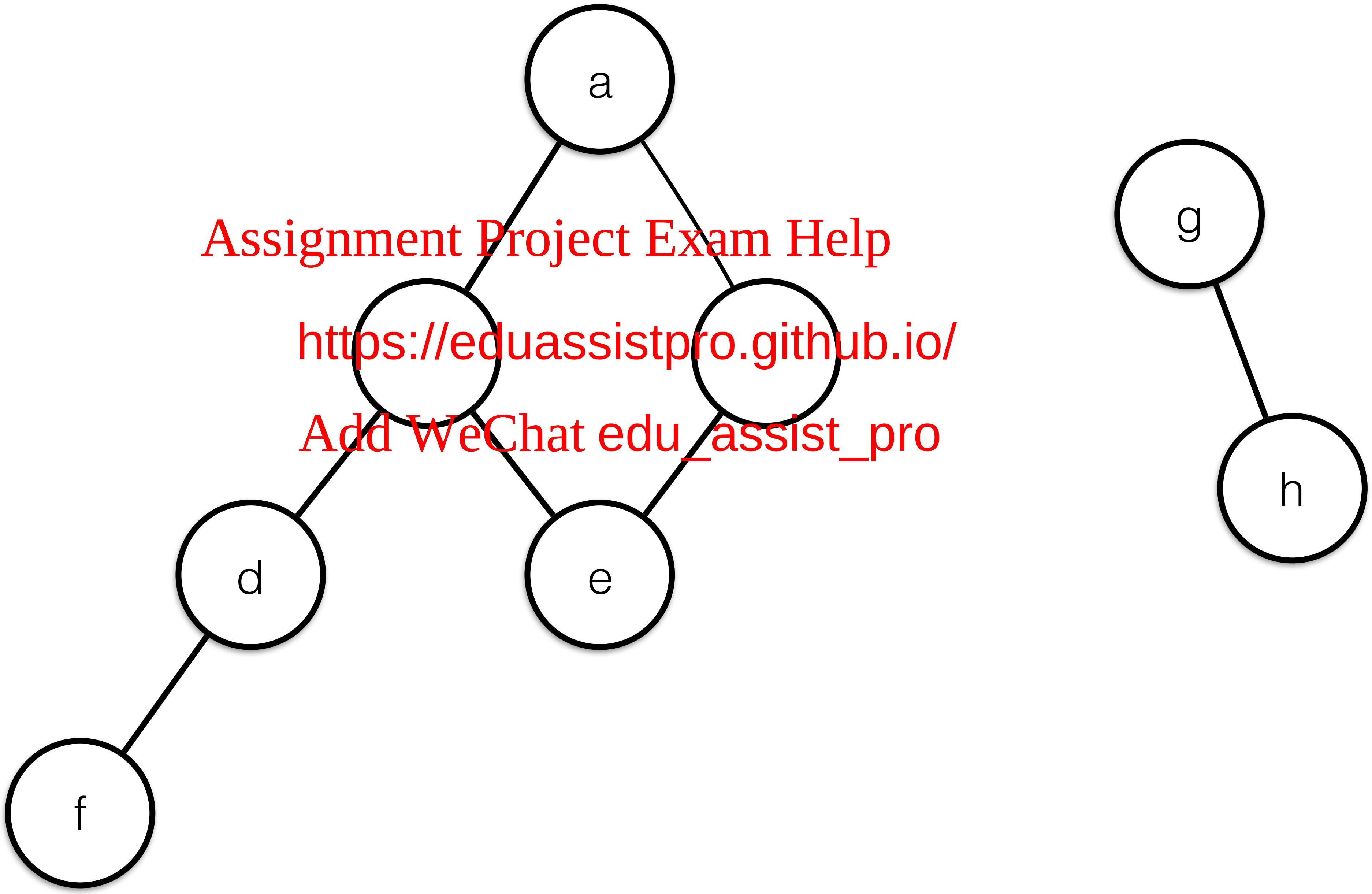
DMD 8.17 (Level 8, Doug McDonell Bldg)

http://people.eng.unimelb.edu.au/tobym

@tobycmurray

Breadth-First and Depth-First Traversal

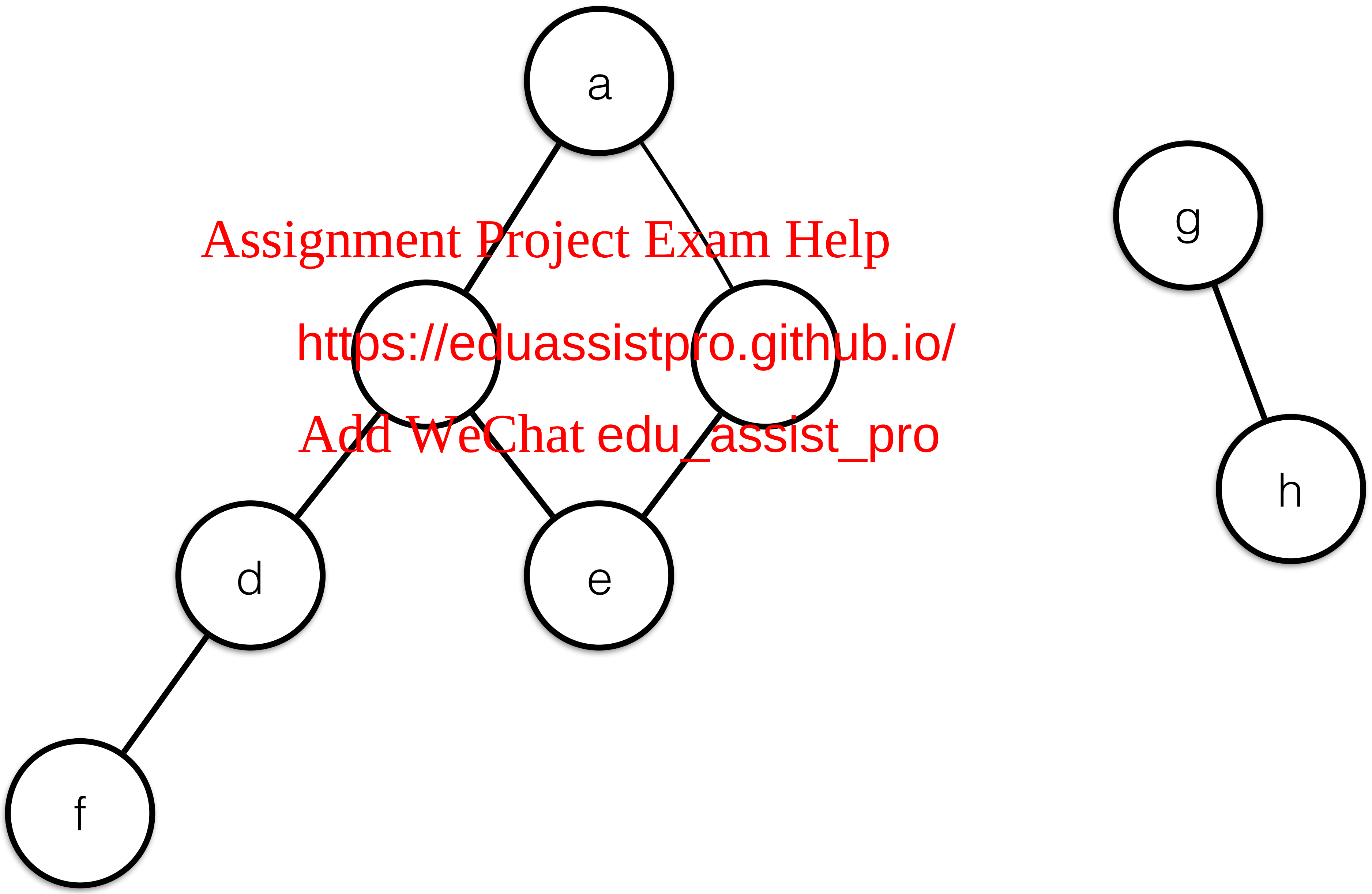
- Used to **systematically** explore **all** nodes of a graph



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal



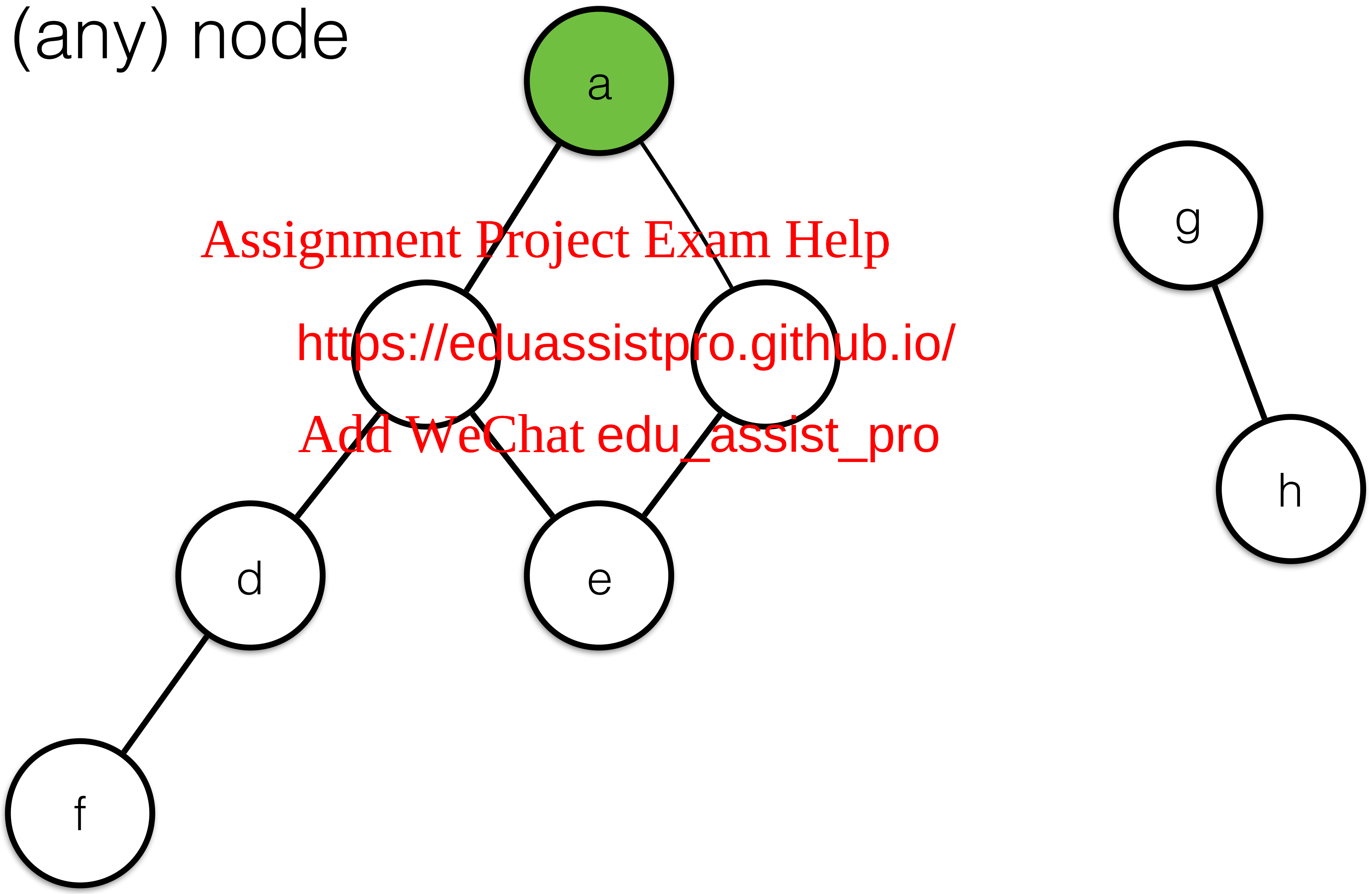
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal

Nodes visited in this order: a

Start with (any) node



Assignment Project Exam Help

https://eduassistpro.github.io/

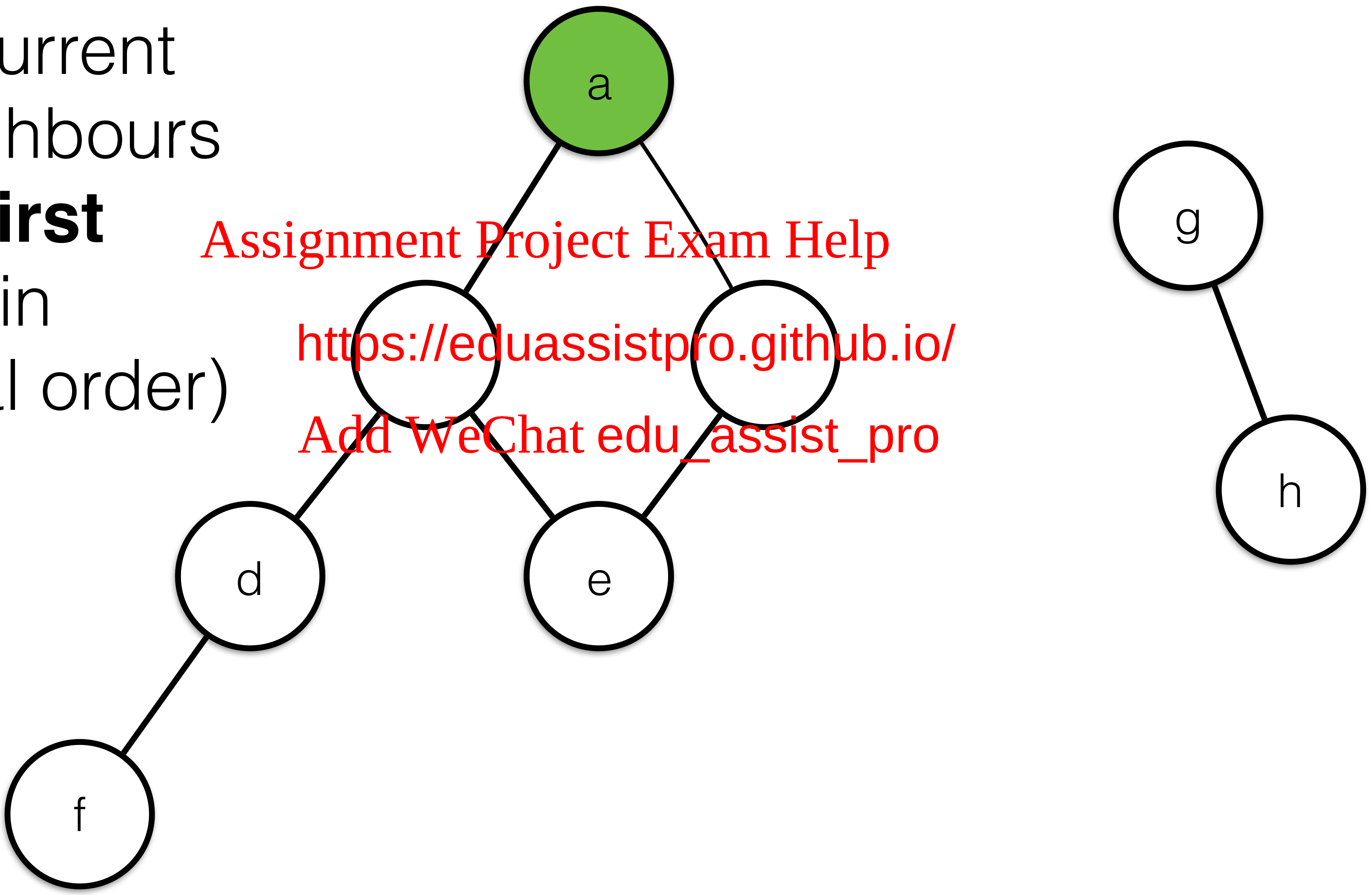
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal

Nodes visited in this order: a

Visit the current node's neighbours **depth first** (here in alphabetical order)



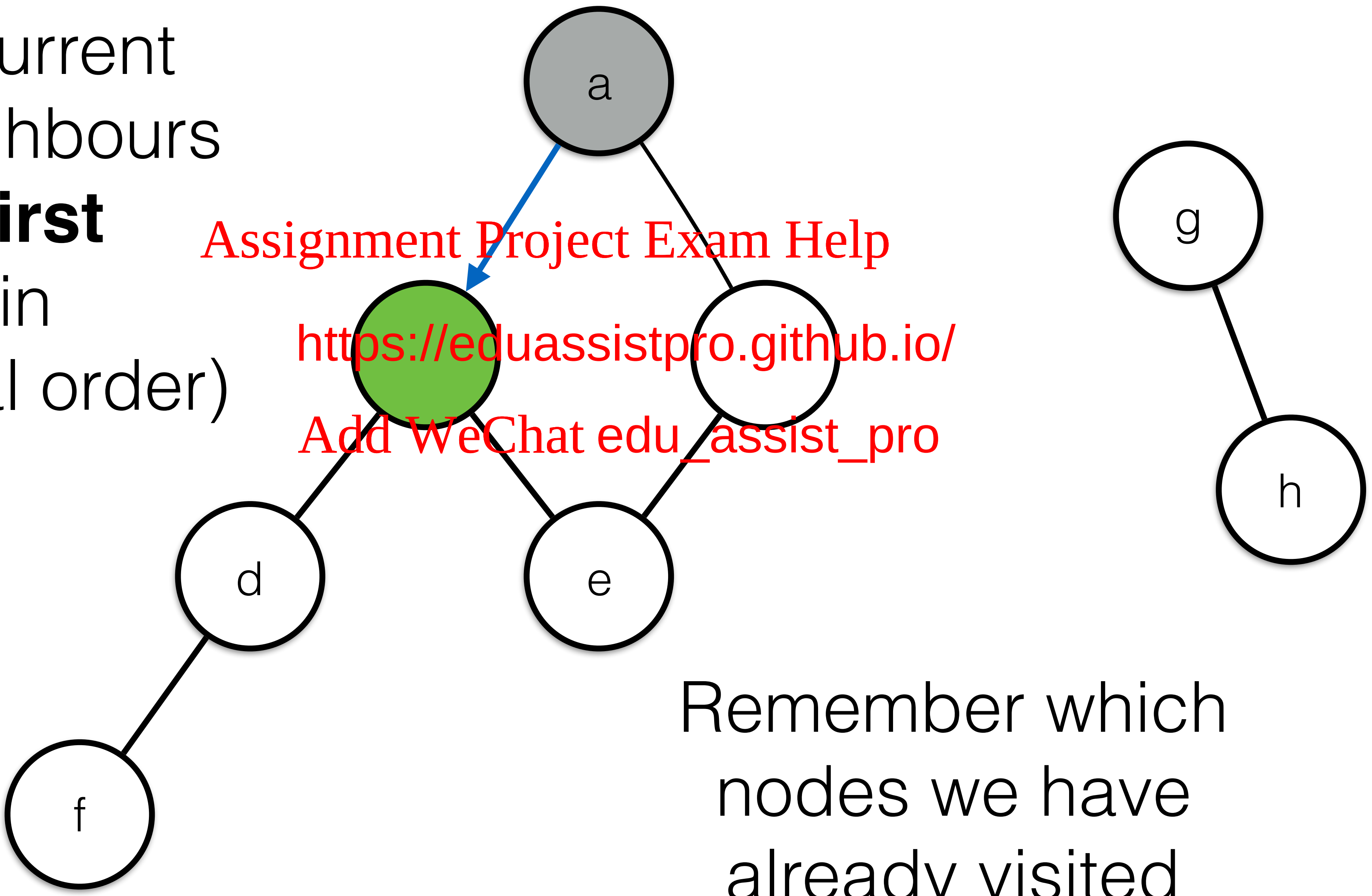
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal

Nodes visited in this order: a b

Visit the current node's neighbours **depth first** (here in alphabetical order)



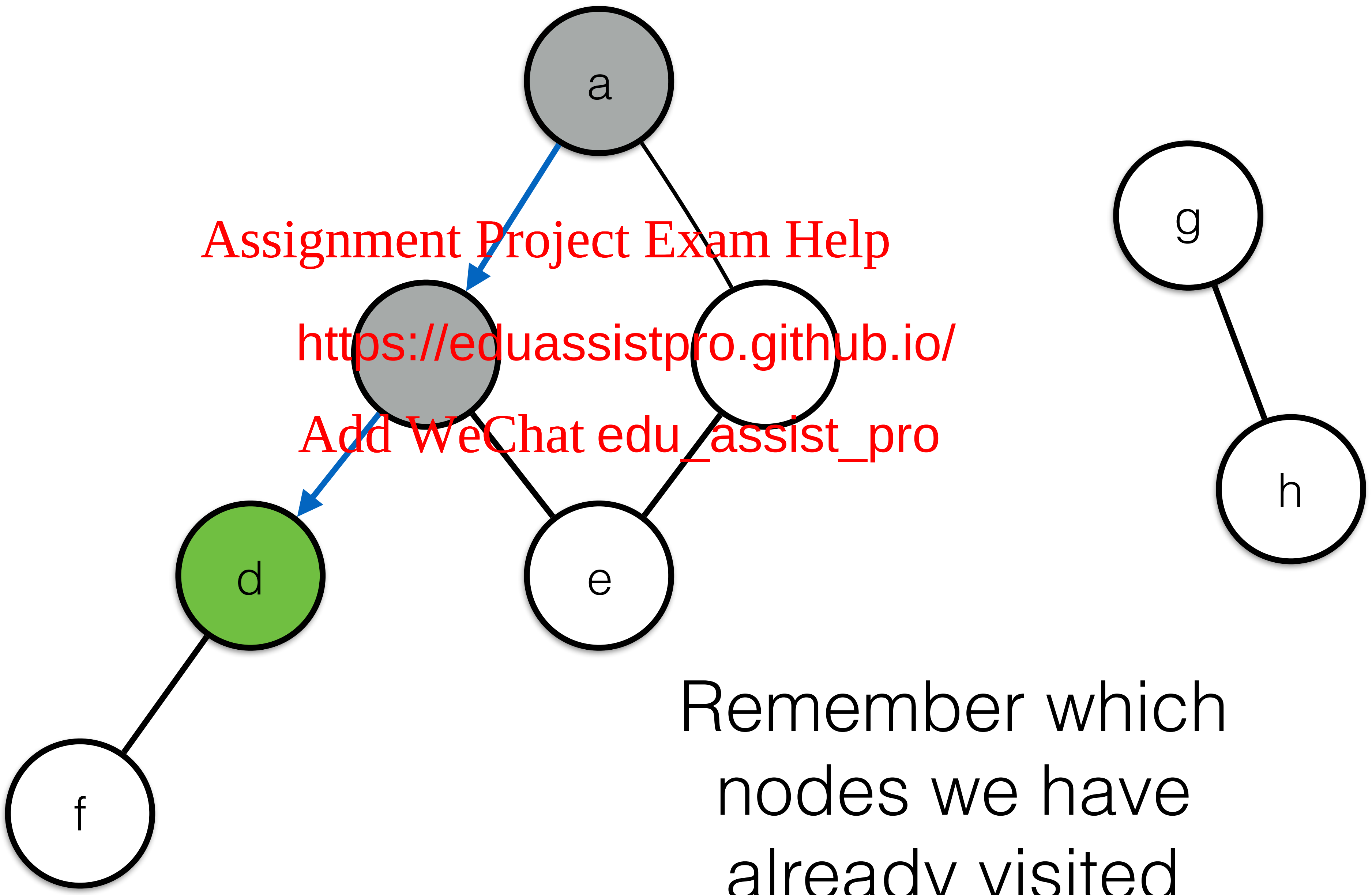
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Remember which nodes we have already visited to avoid revisiting them

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal

Nodes visited in this order: a b d



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

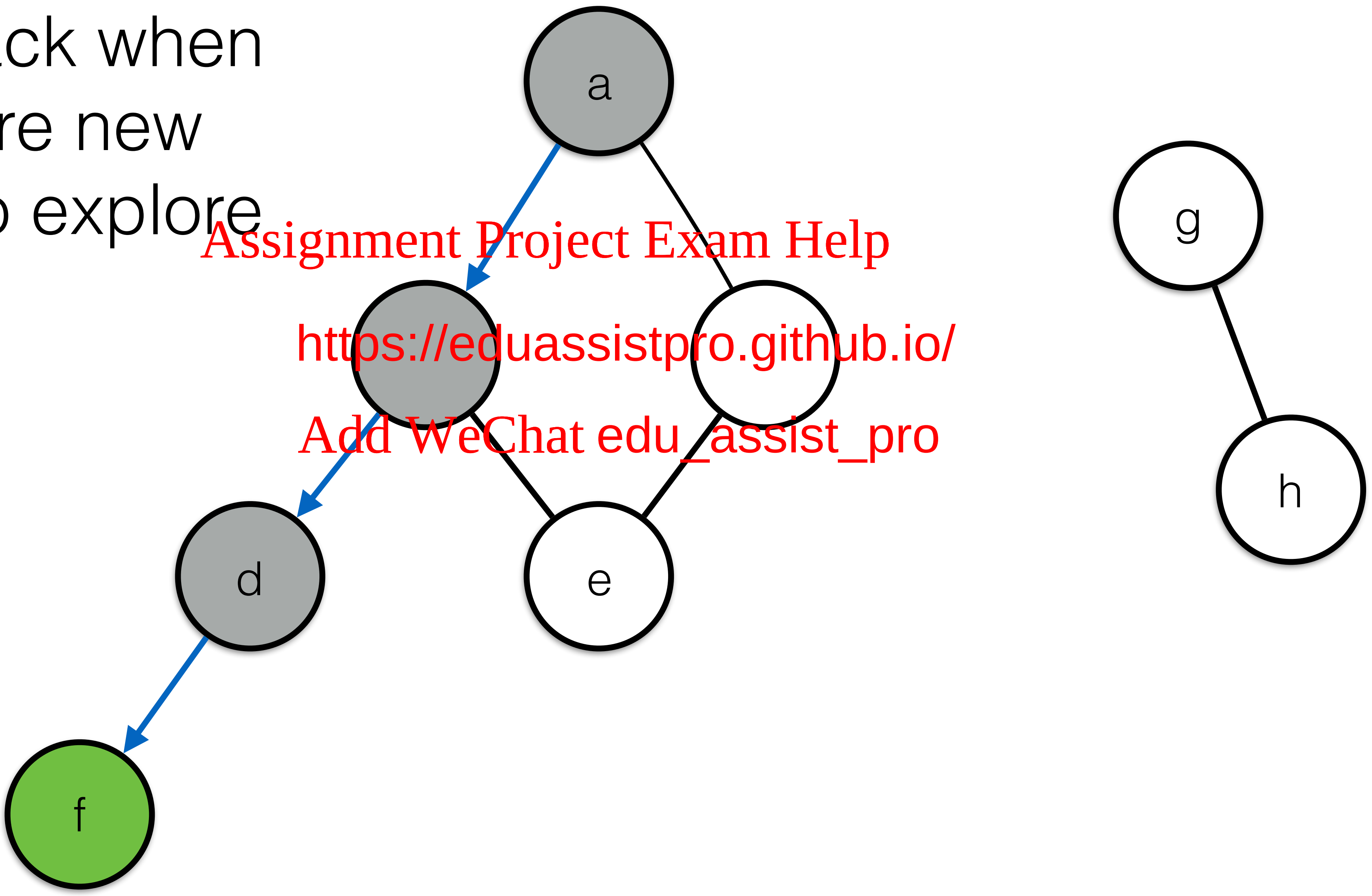
Remember which nodes we have already visited to avoid revisiting them

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal

Nodes visited in this order: a b d f

Back-track when no more new nodes to explore



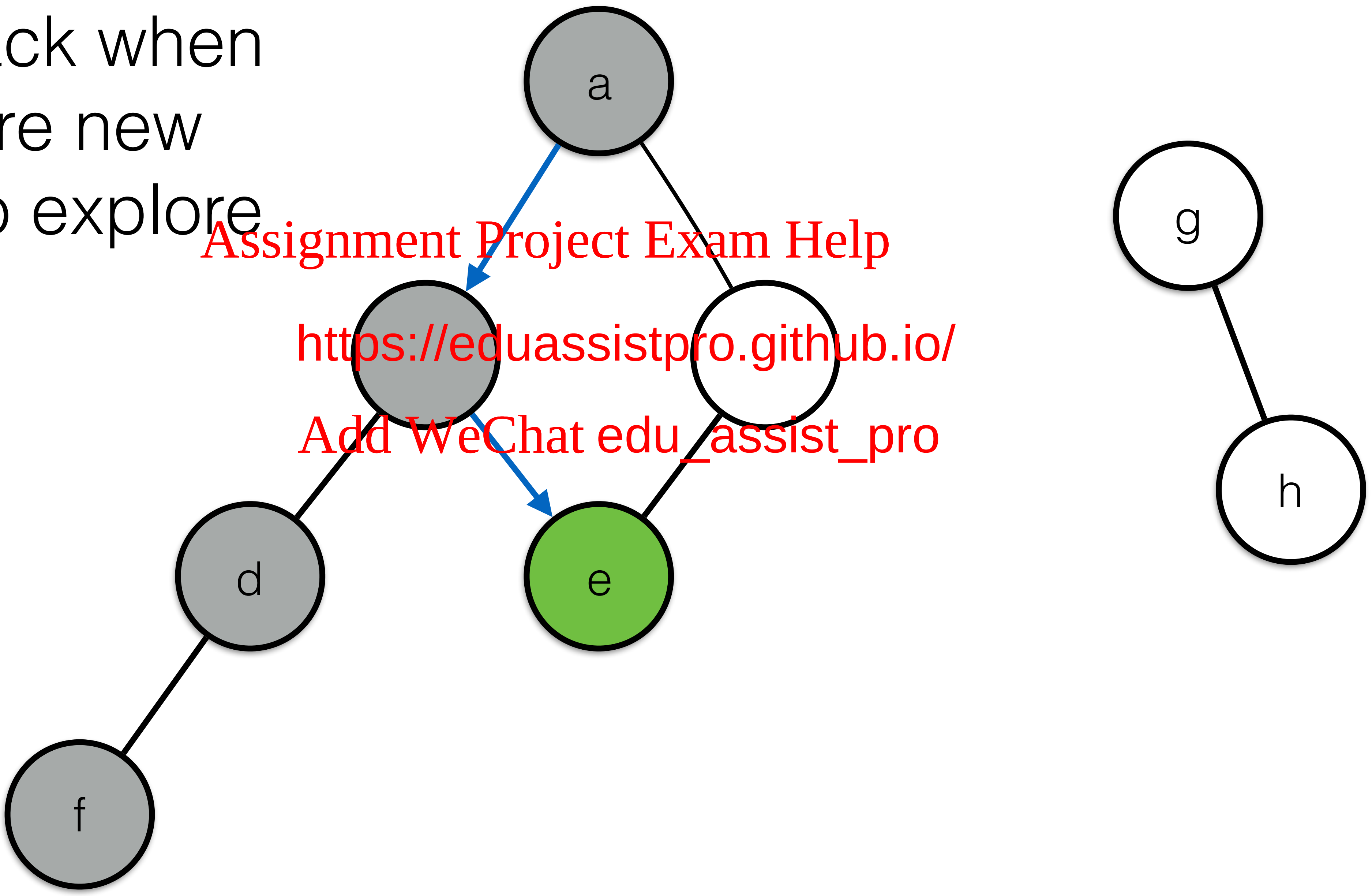
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal

Nodes visited in this order: a b d f e

Back-track when no more new nodes to explore



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

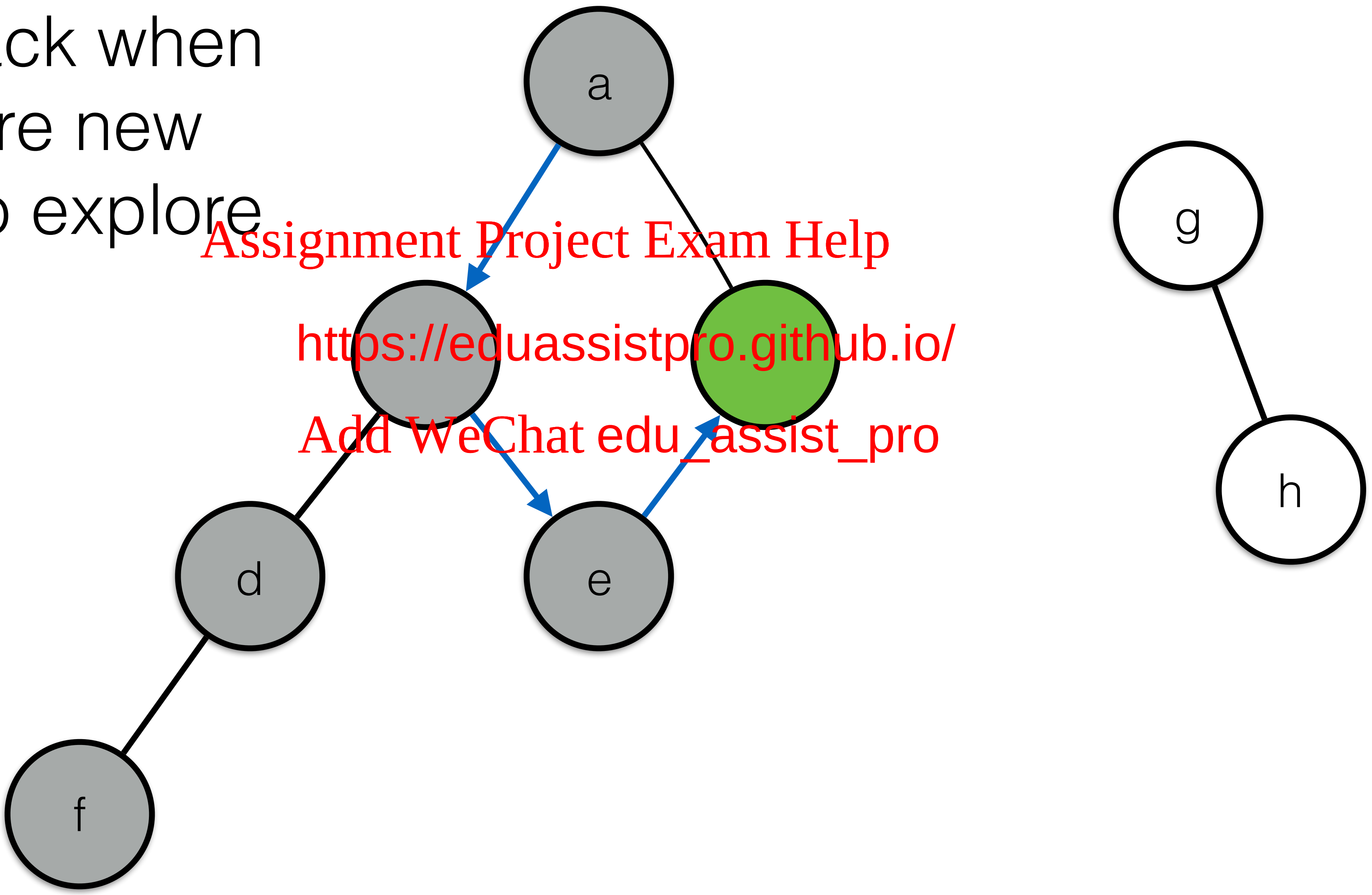
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal

Nodes visited in this order: a b d f e c

Back-track when no more new nodes to explore

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

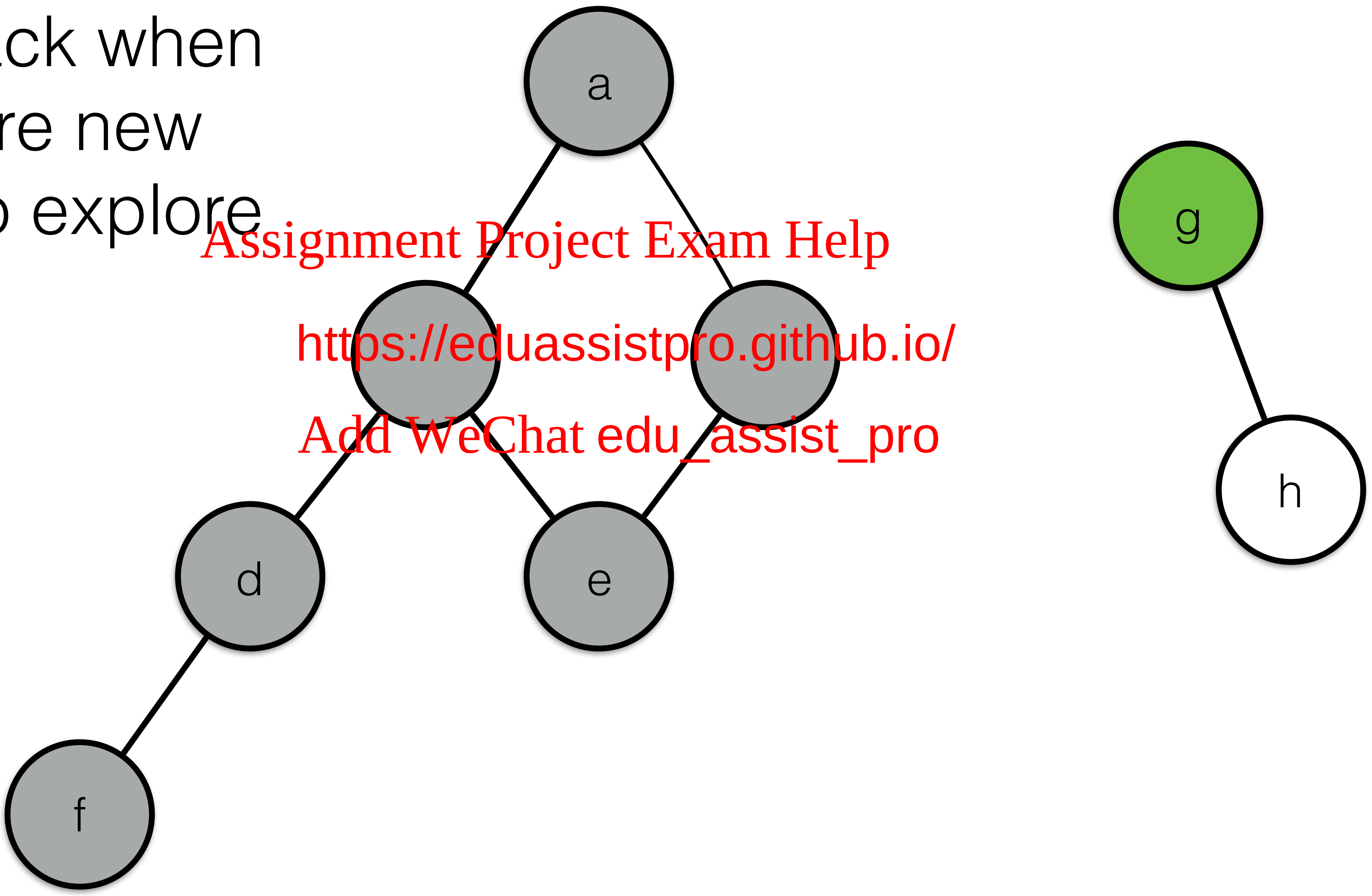


Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal

Nodes visited in this order: a b d f e c g

Back-track when no more new nodes to explore

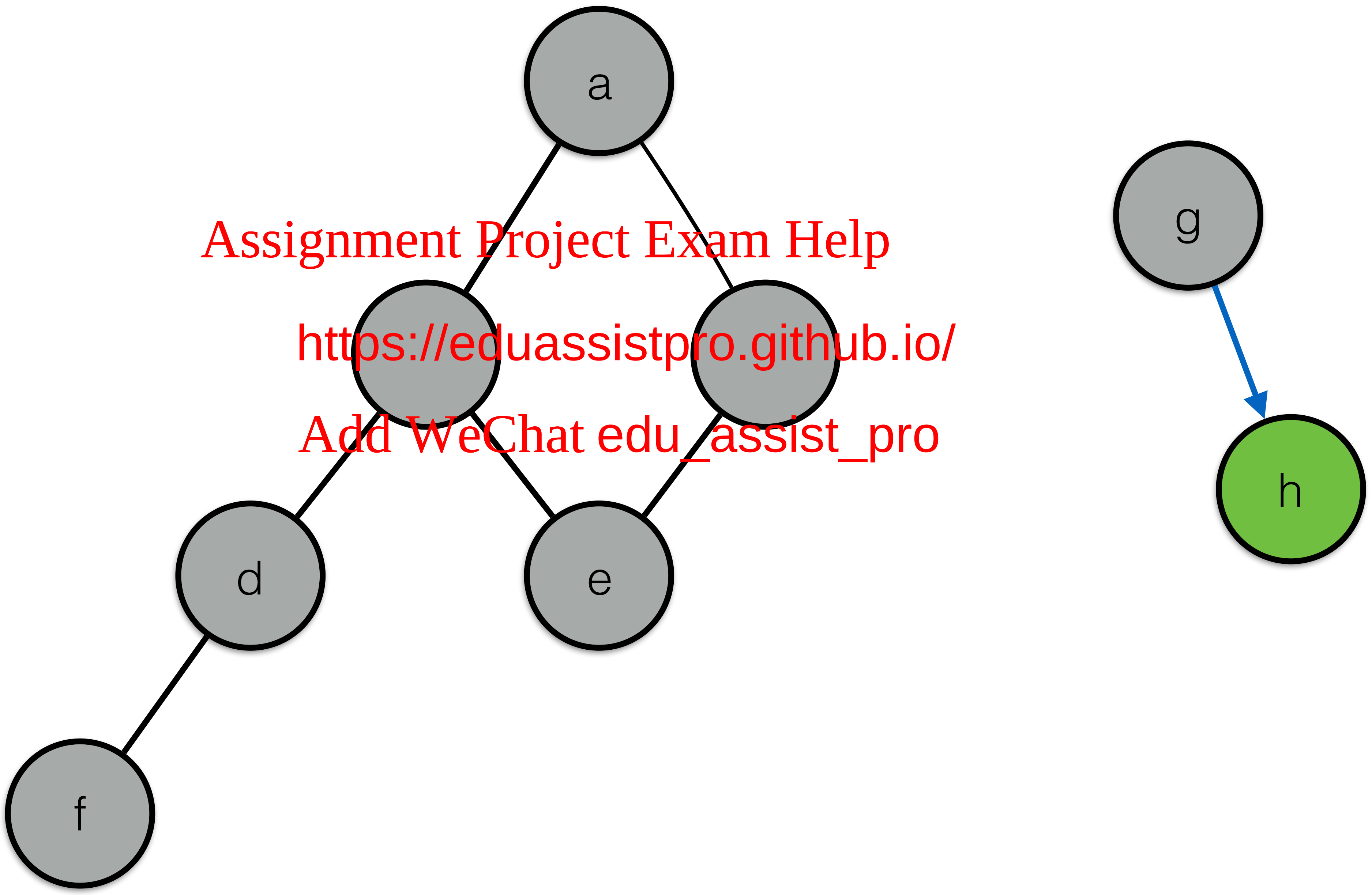


Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal

Nodes visited in this order: a b d f e c g h



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Stack Discipline

When back-tracking, we go back to the most recently-visited node that still has unvisited neighbours

This is simulated by pushing each node onto a stack as it is visited.

Assignment Project Exam Help

https://eduassistpro.github.io/

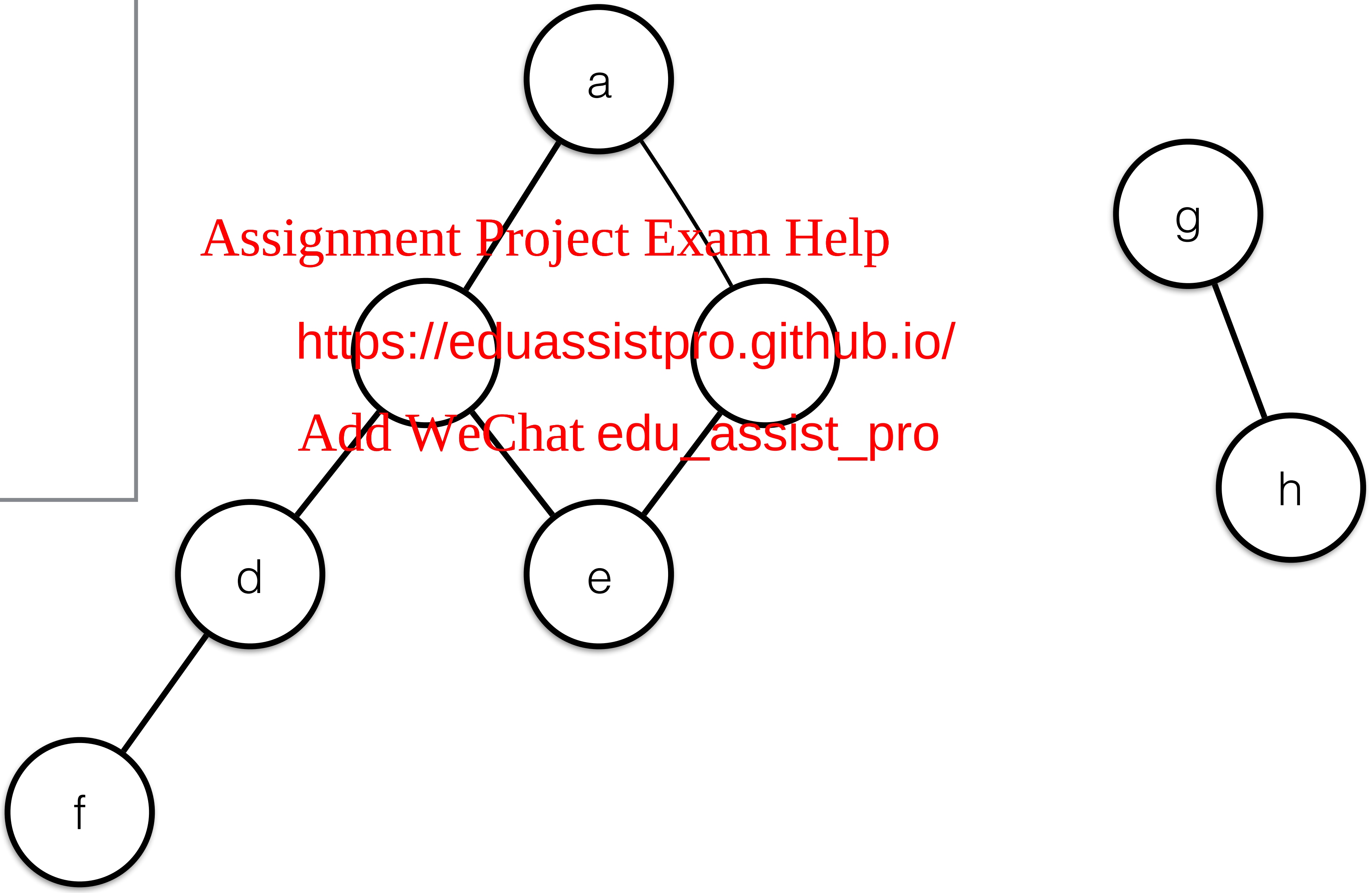
Add WeChat edu_assist_pro

Back-tracking then correspo popping the stack.

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Stack Discipline

(Simulated) stack:



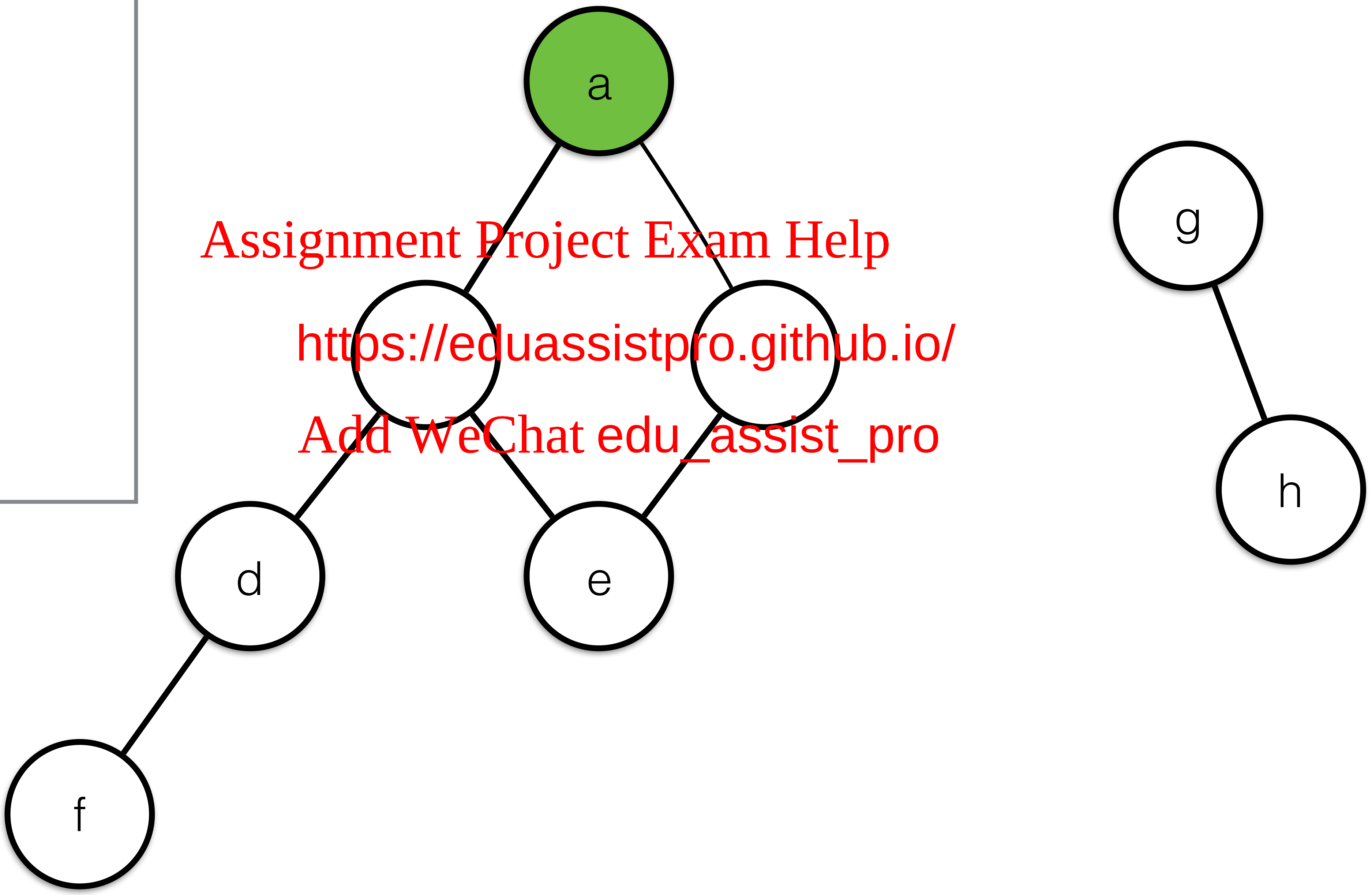
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Stack Discipline

(Simulated) stack:

a



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

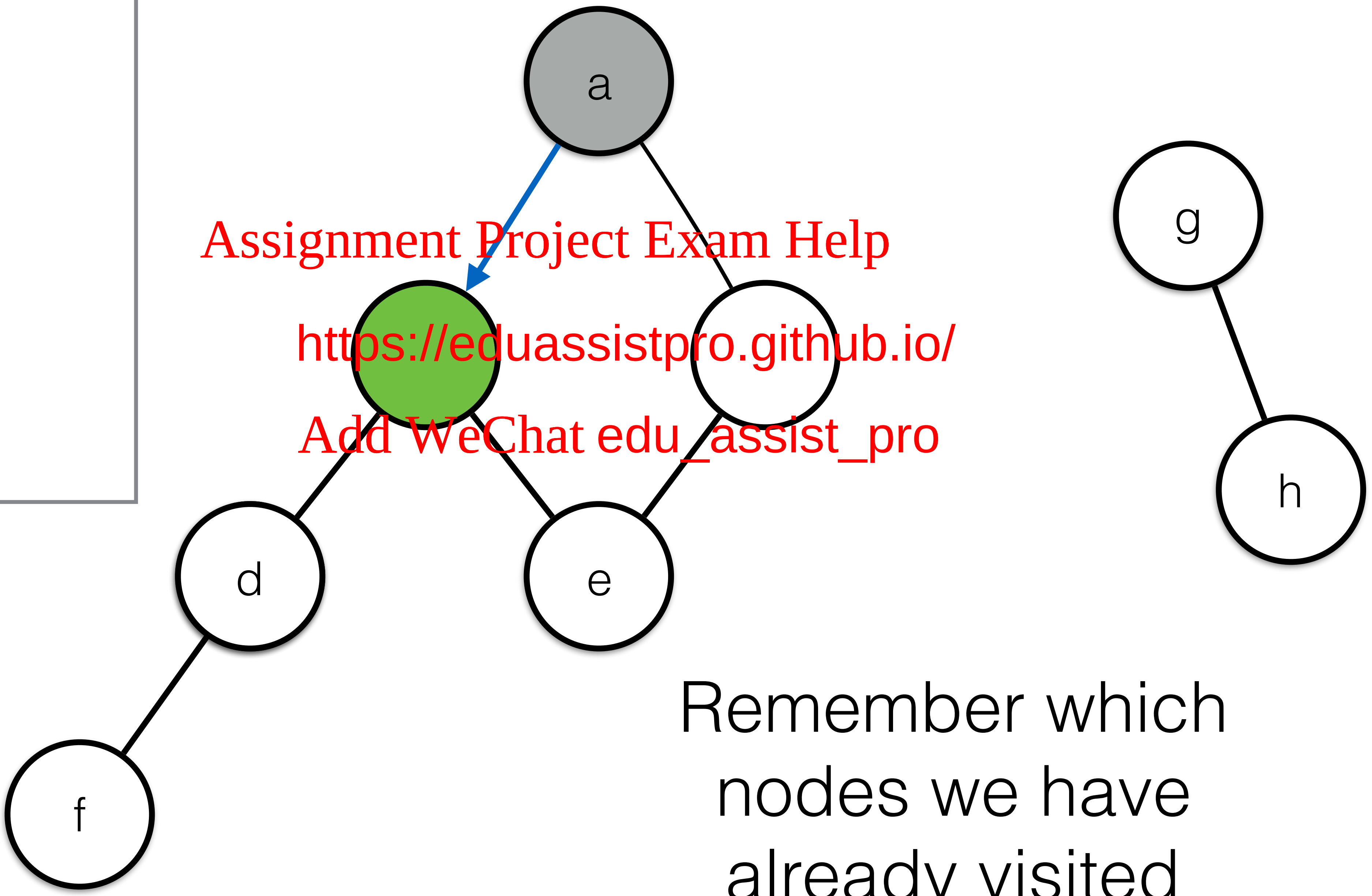
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Stack Discipline

(Simulated) stack:

b
a

a b



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

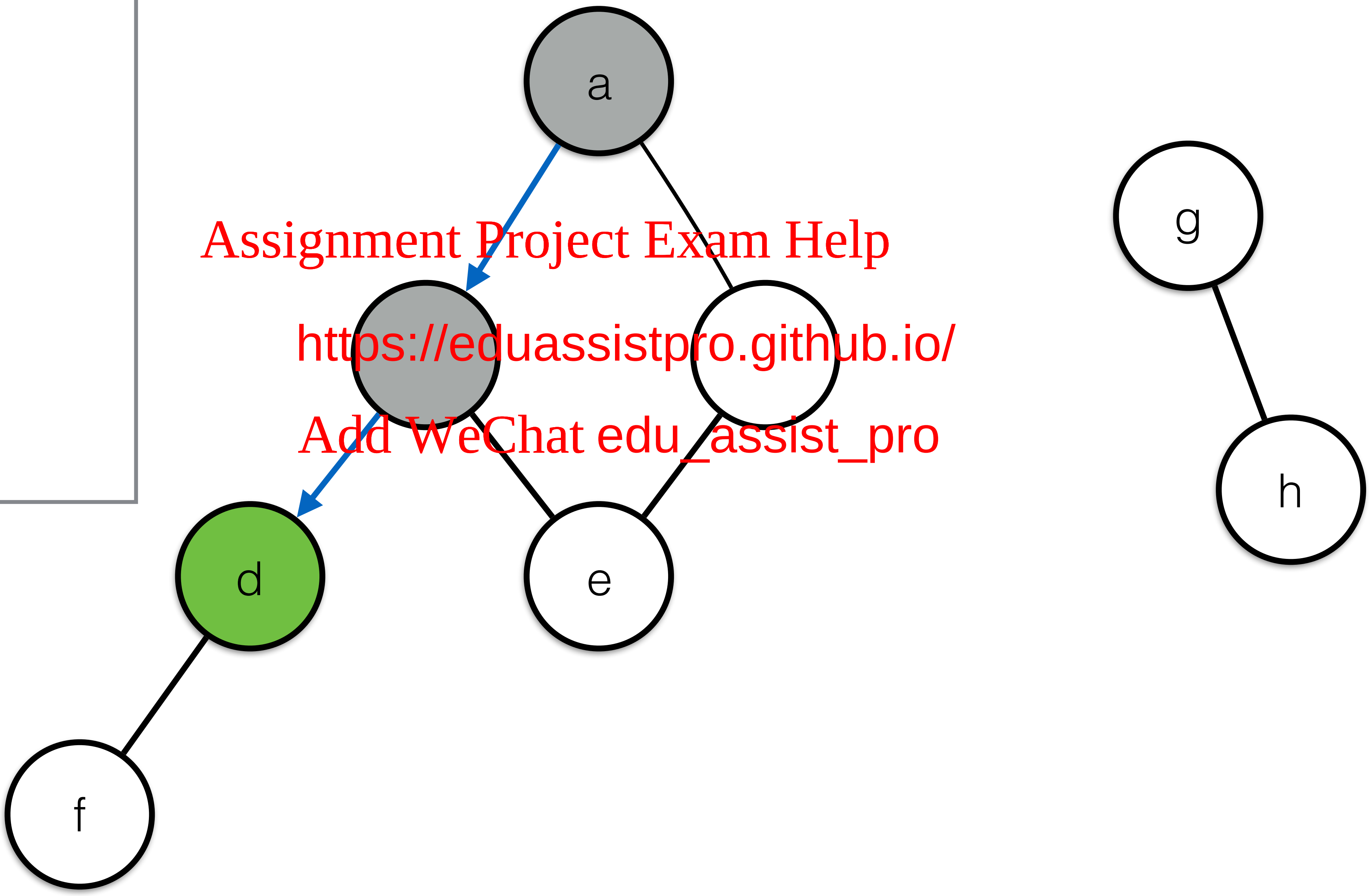
Remember which nodes we have already visited to avoid revisiting them

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Stack Discipline

(Simulated) stack:

d
b
a



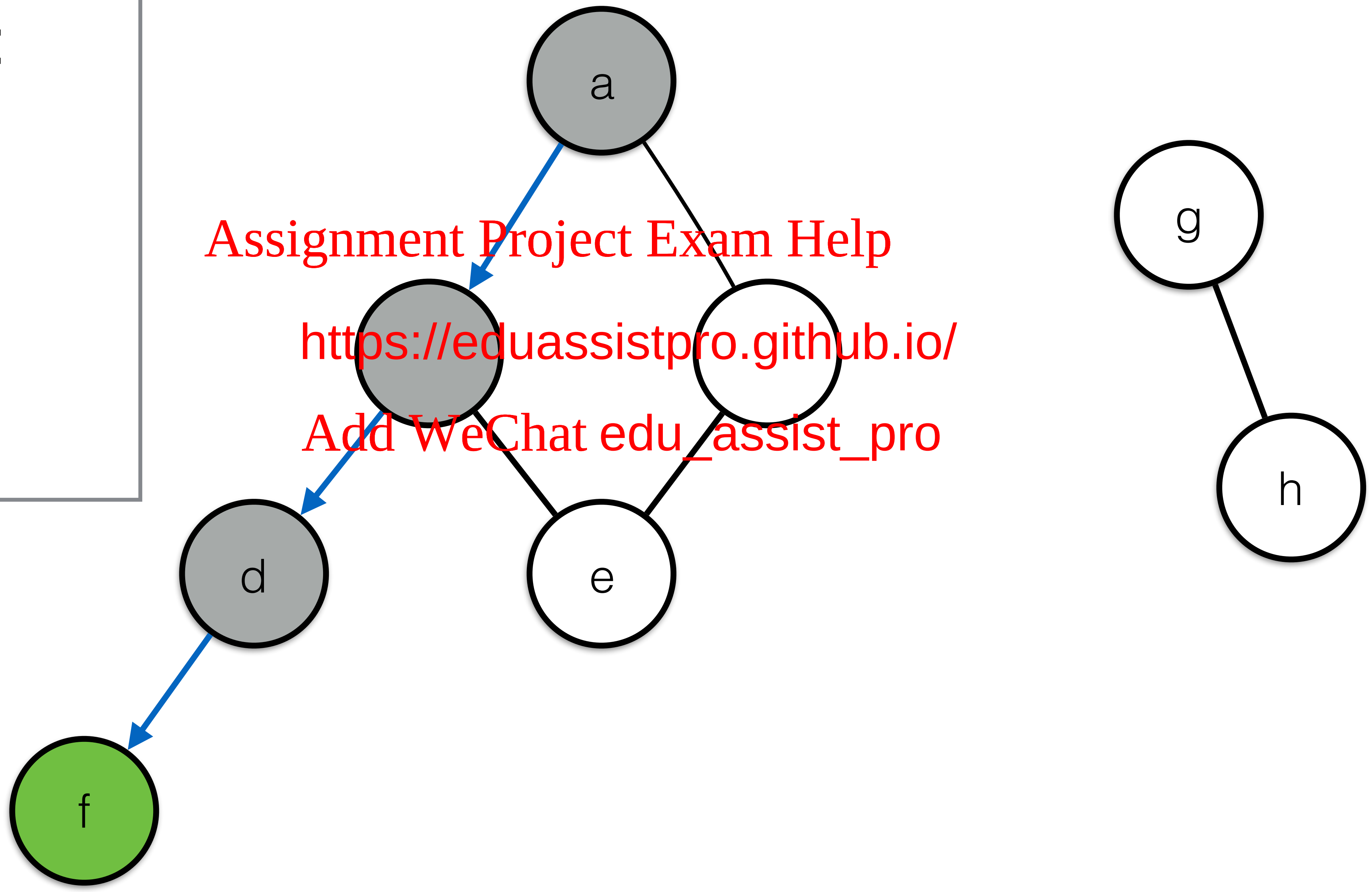
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Stack Discipline

(Simulated) stack:

f
d
b
a



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

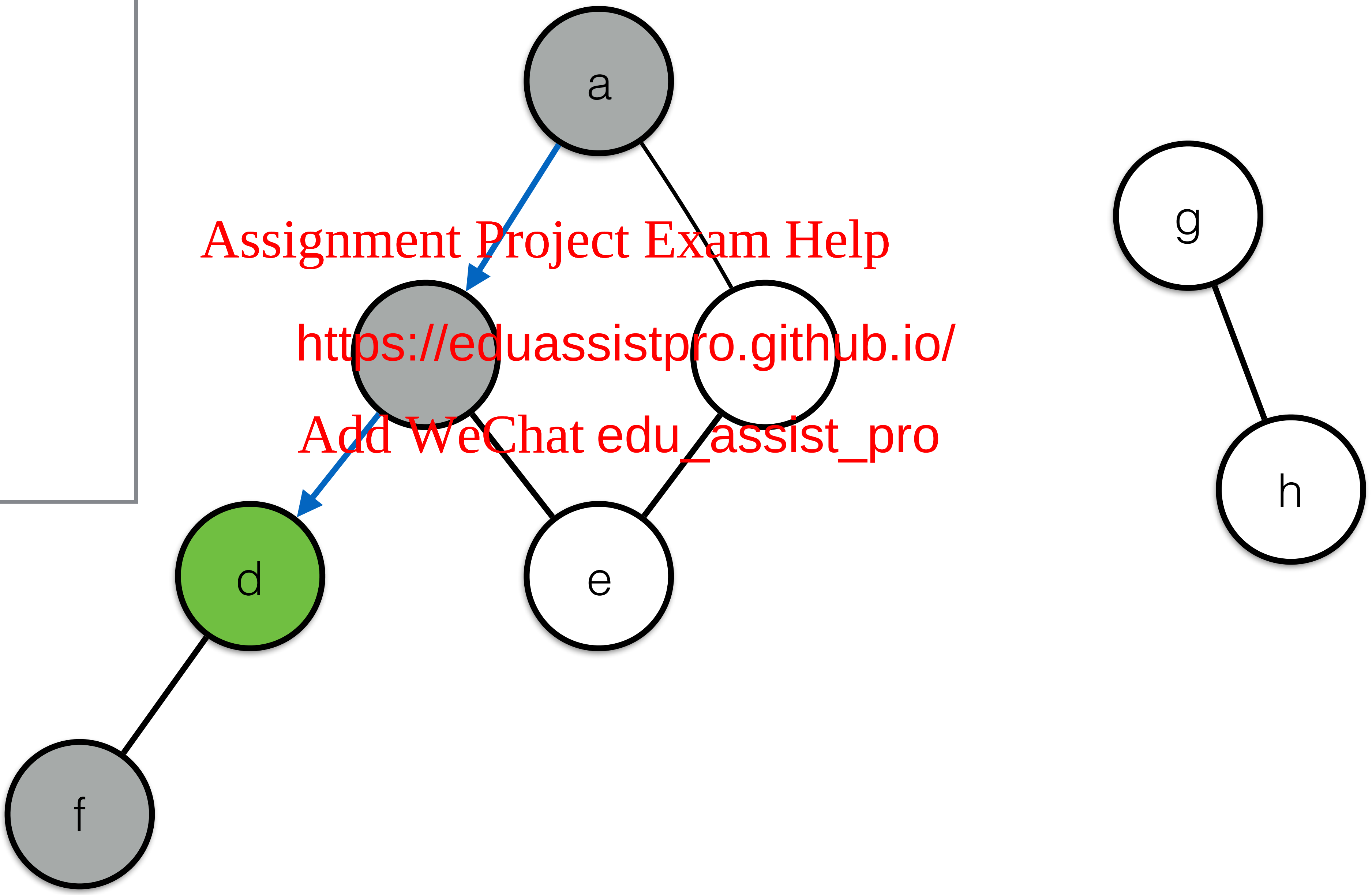
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Stack Discipline

(Simulated) stack:

d
b
a

(back-tracking)



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

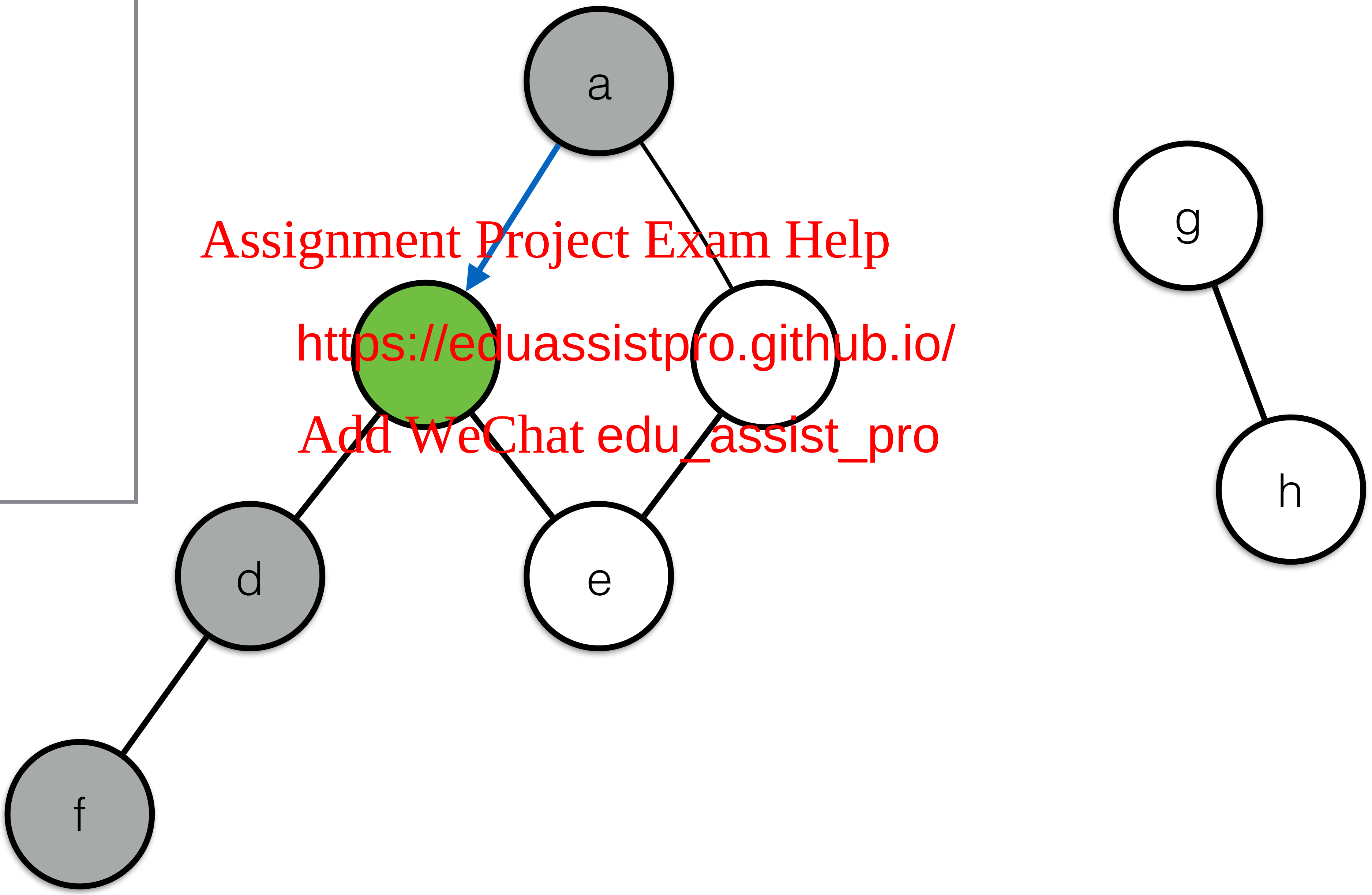
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Stack Discipline

(Simulated) stack:

b
a

(back-tracking)



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

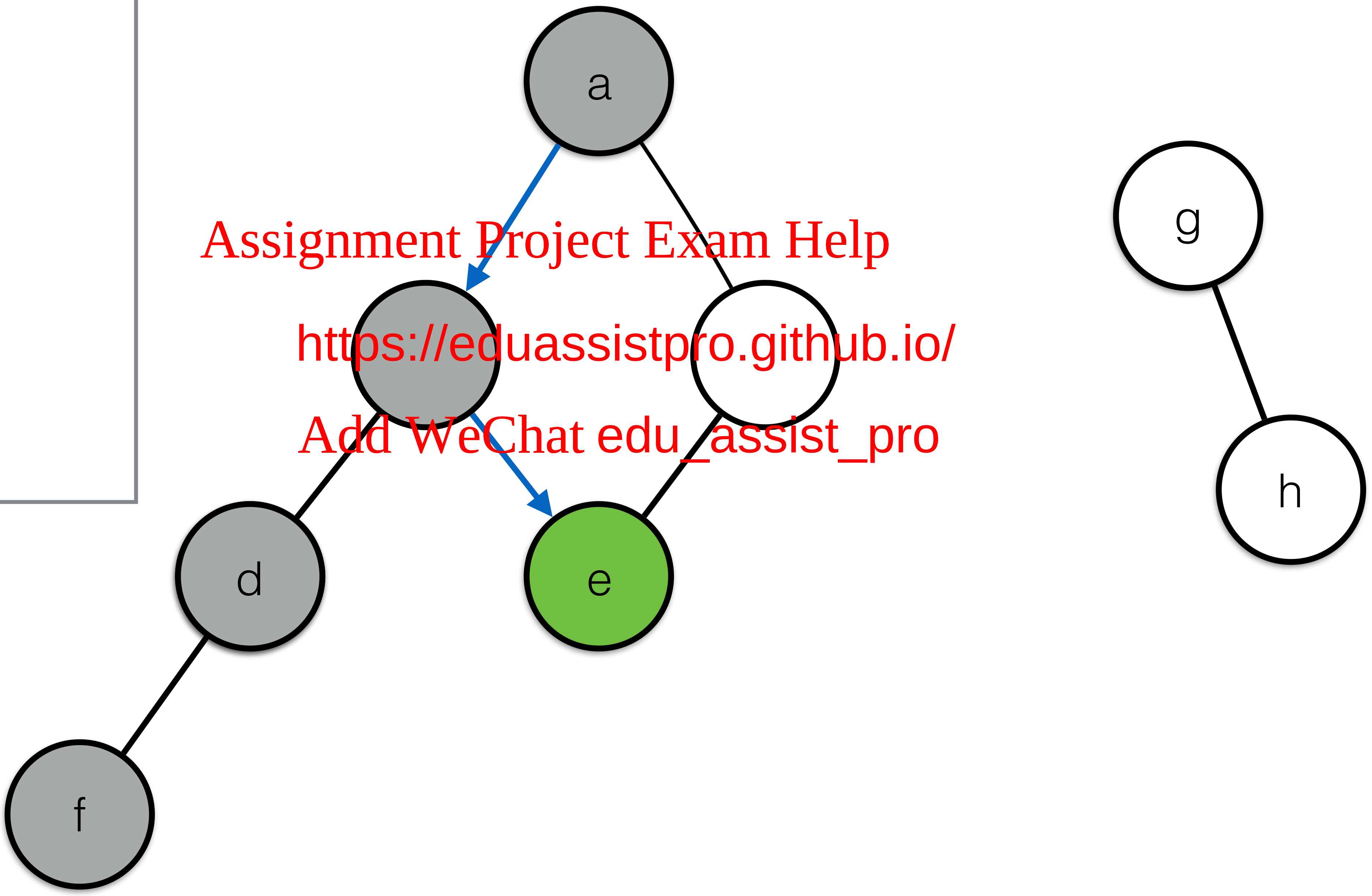
Depth-First Traversal: Stack Discipline

(Simulated) stack:

e

b

a



Assignment Project Exam Help

https://eduassistpro.github.io/

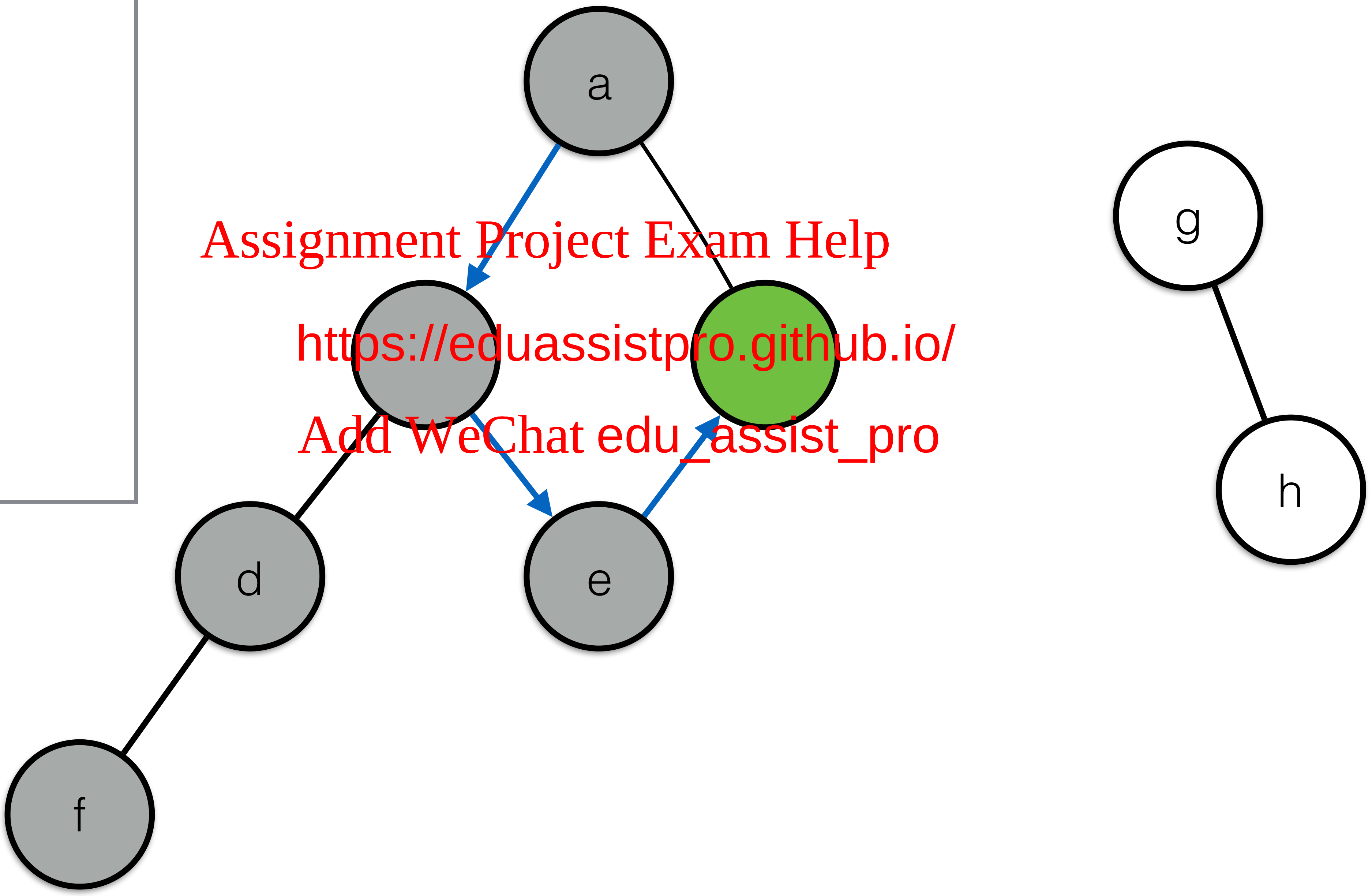
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Stack Discipline

(Simulated) stack:

c
e
b
a



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

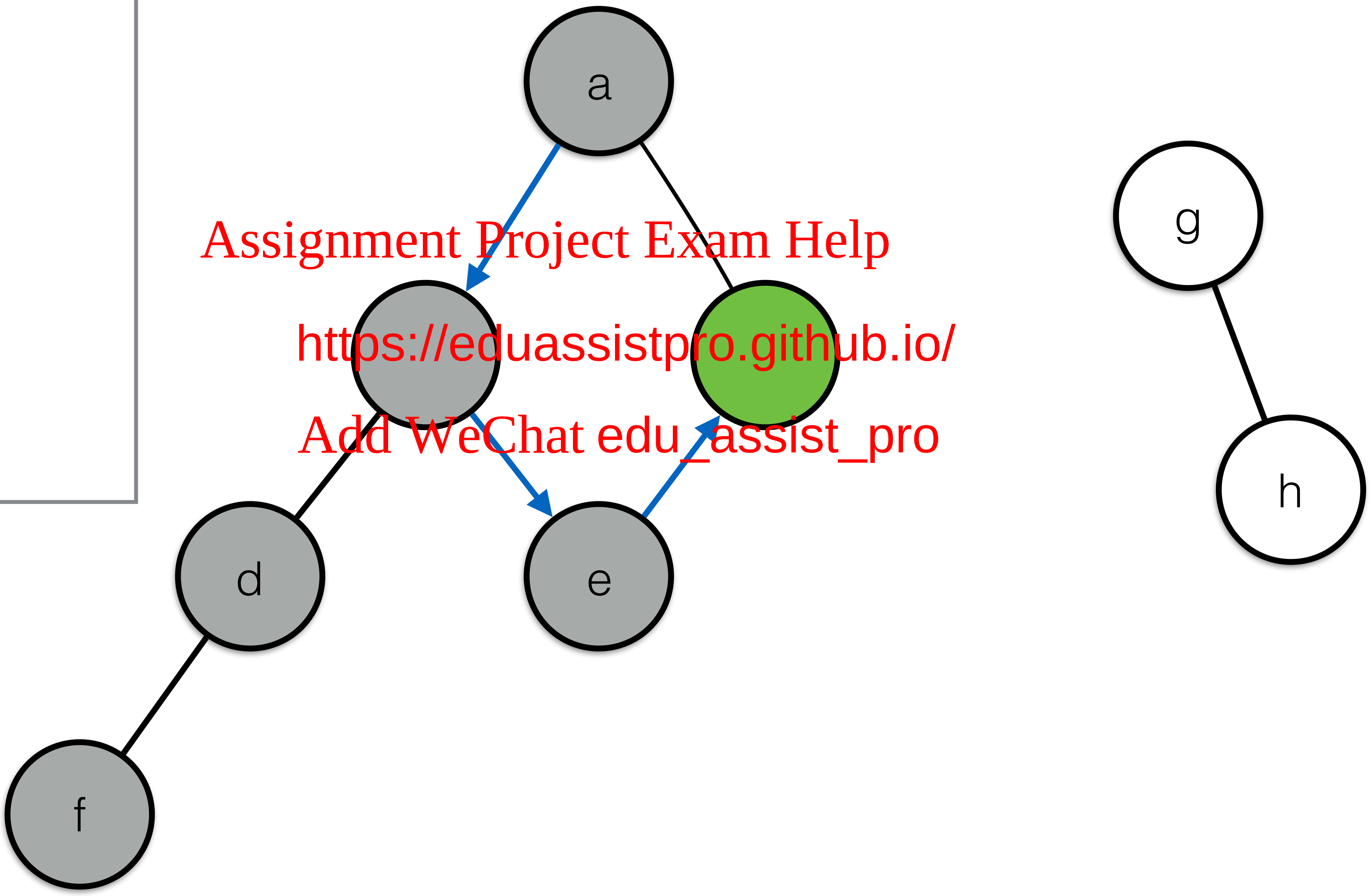
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Stack Discipline

(Simulated) stack:

c
e
b
a

back-tracking …
details omitted …

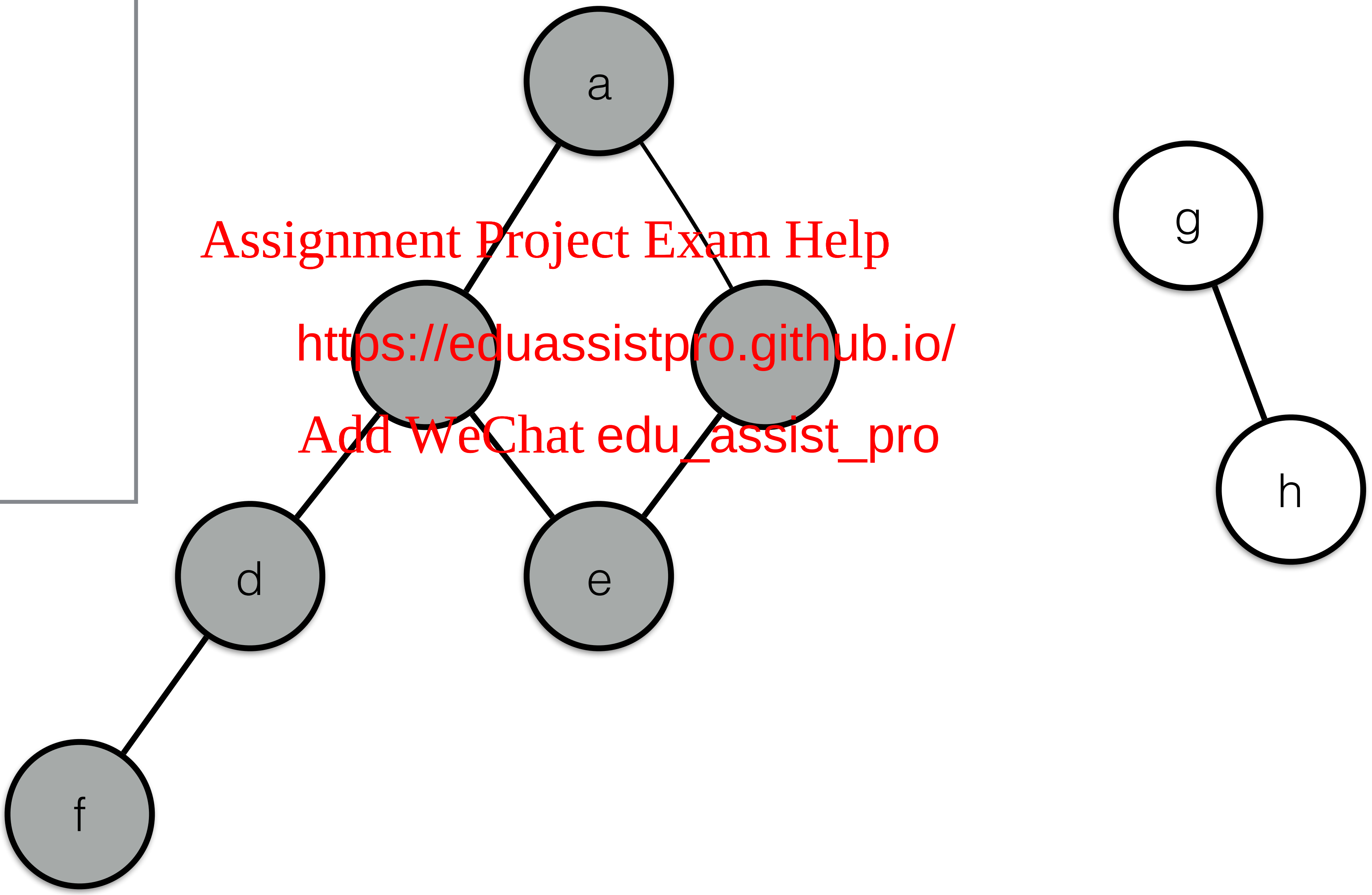


Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Stack Discipline

(Simulated) stack:



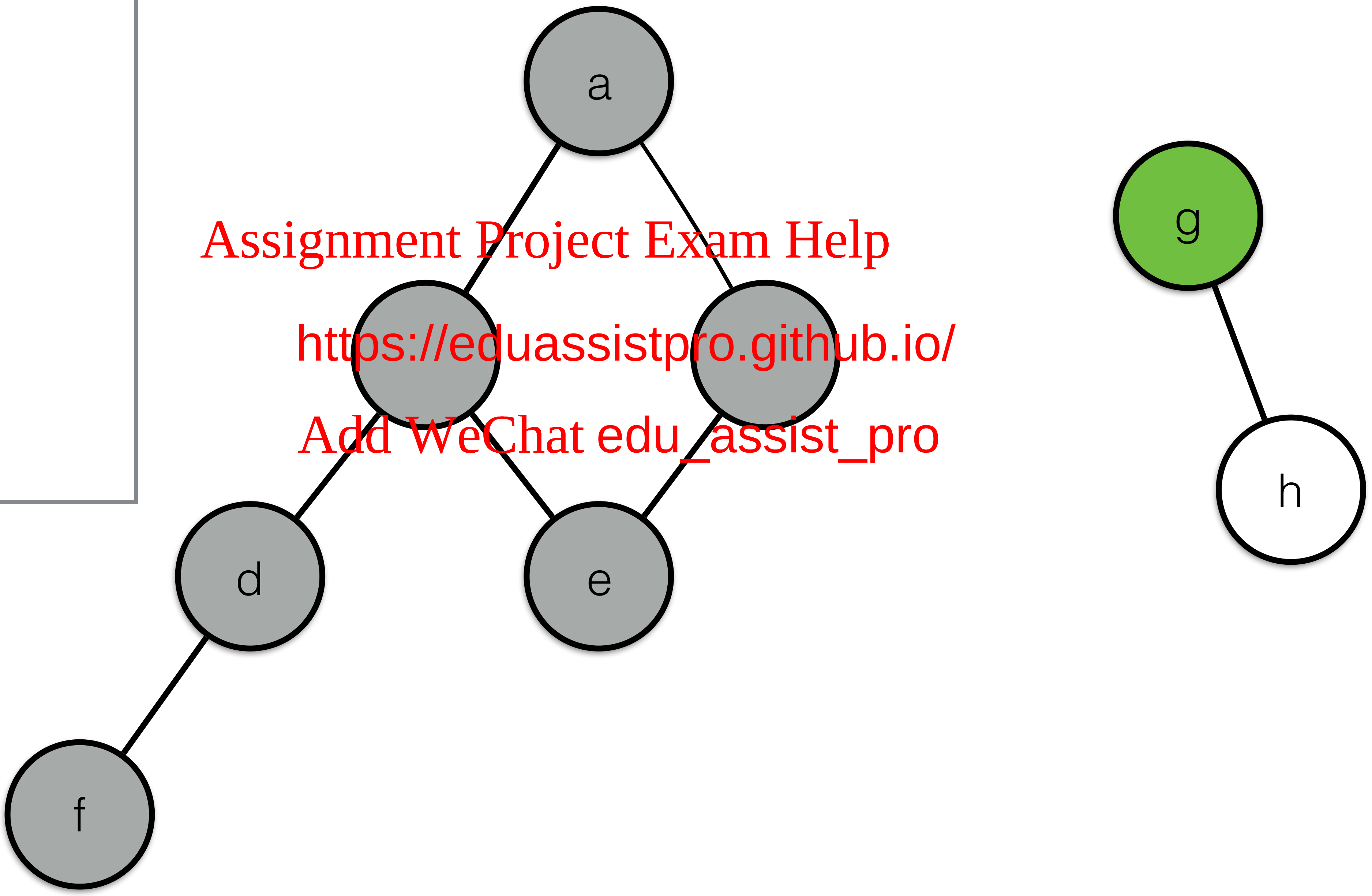
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Stack Discipline

(Simulated) stack:

g



Assignment Project Exam Help

https://eduassistpro.github.io/

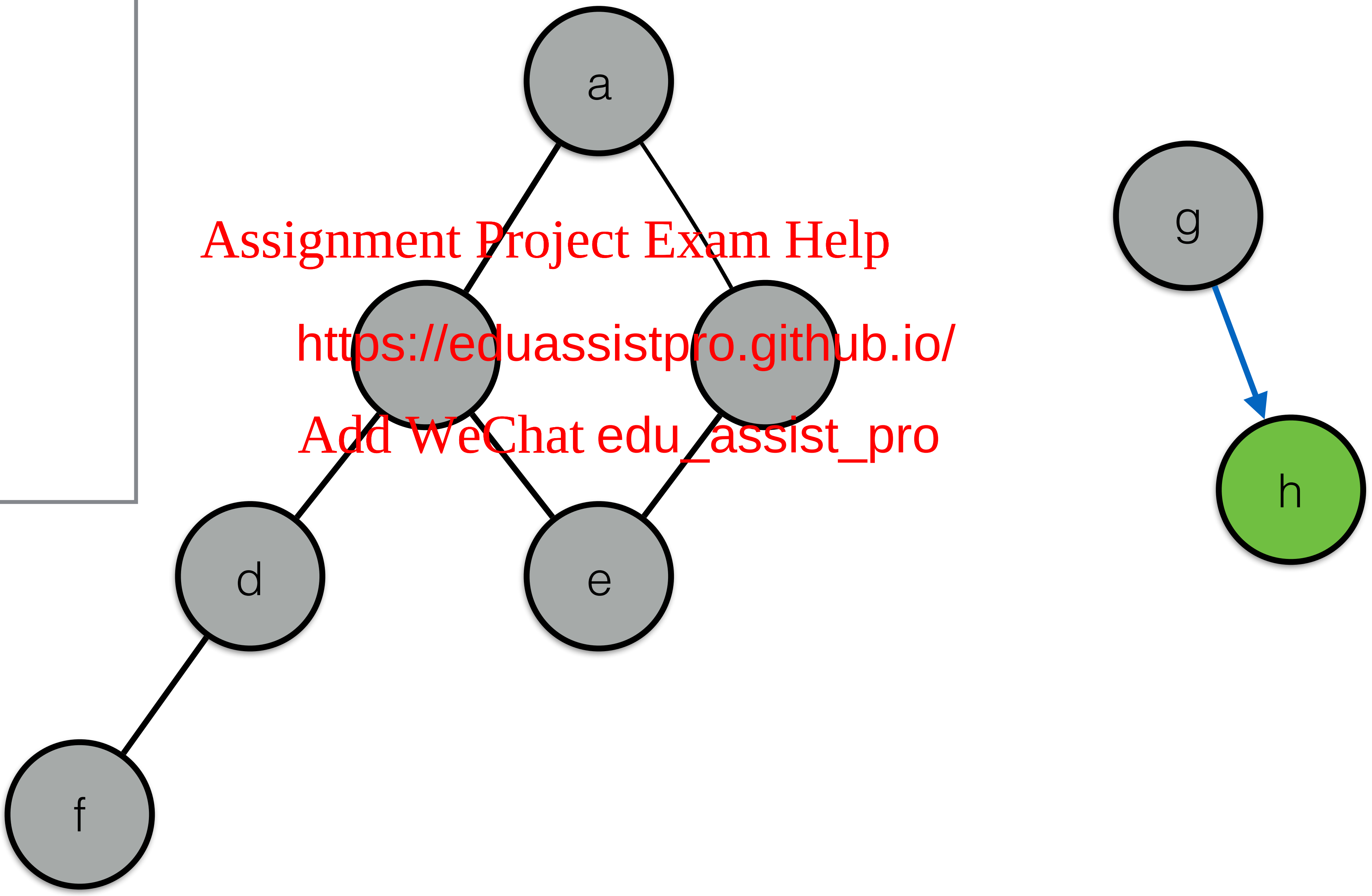
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Stack Discipline

(Simulated) stack:

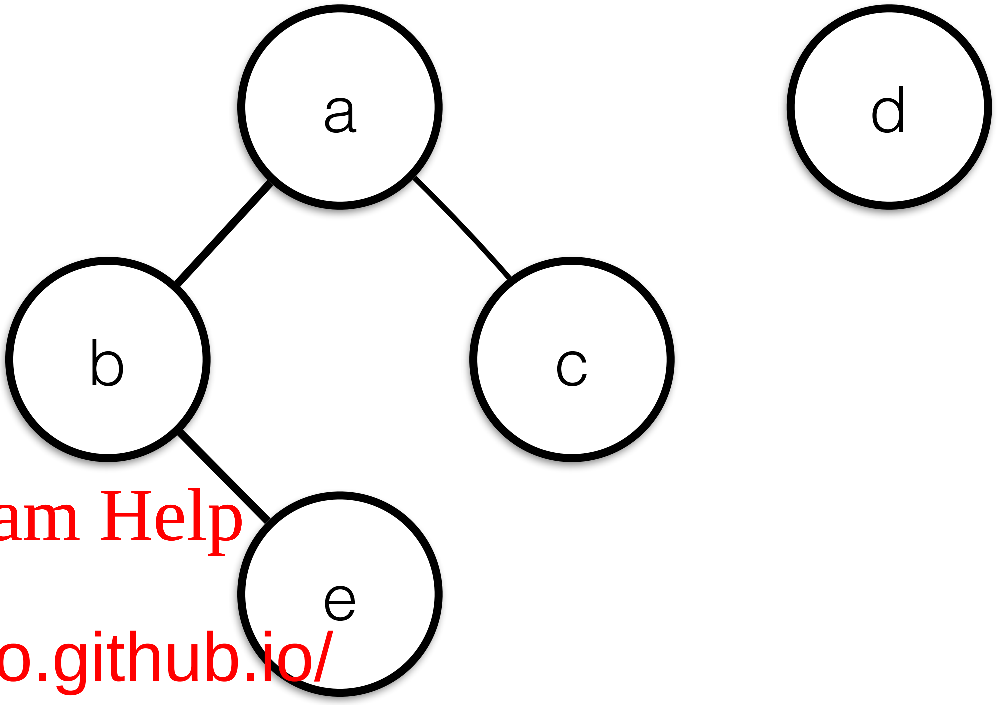
h
g



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

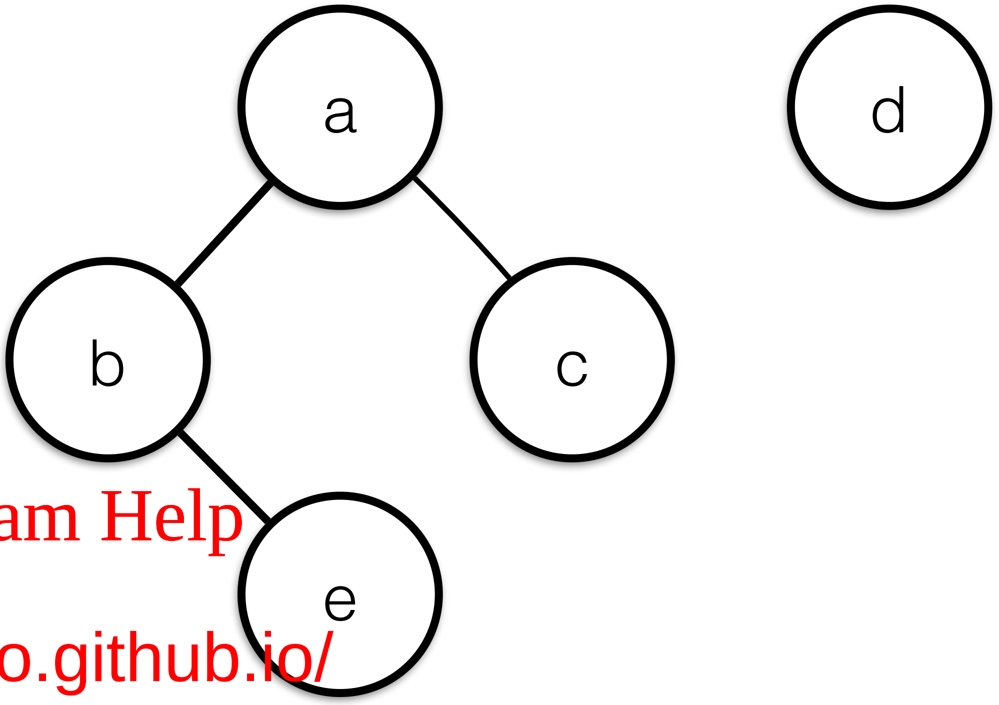
https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Call Stack

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

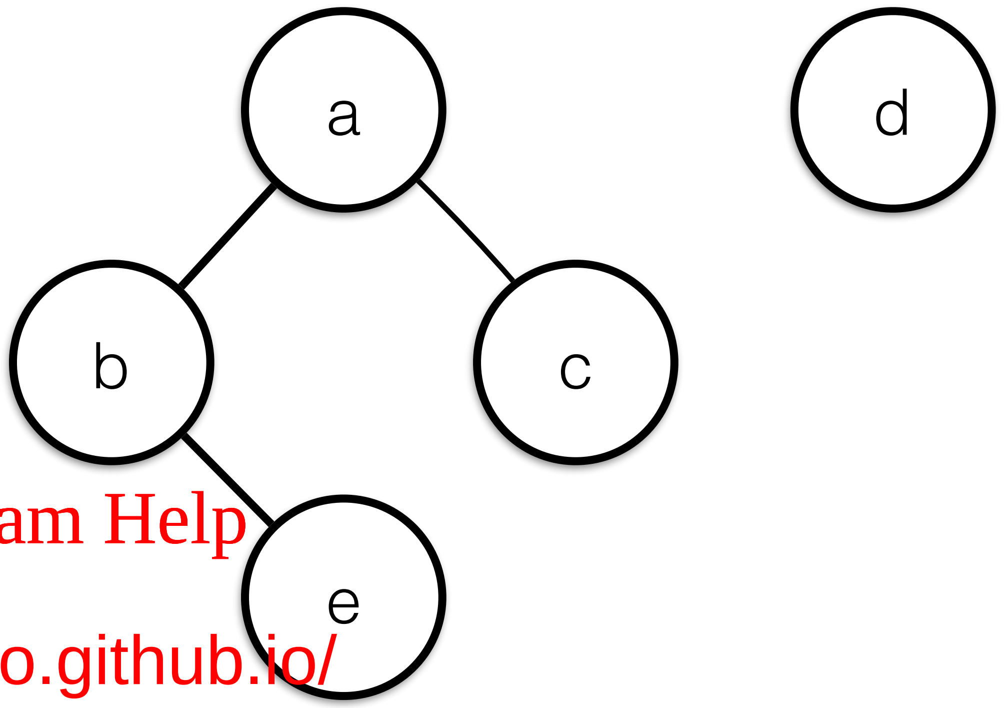
Add WeChat edu_assist_pro

DFS(⟨V,E⟩)

Call Stack

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

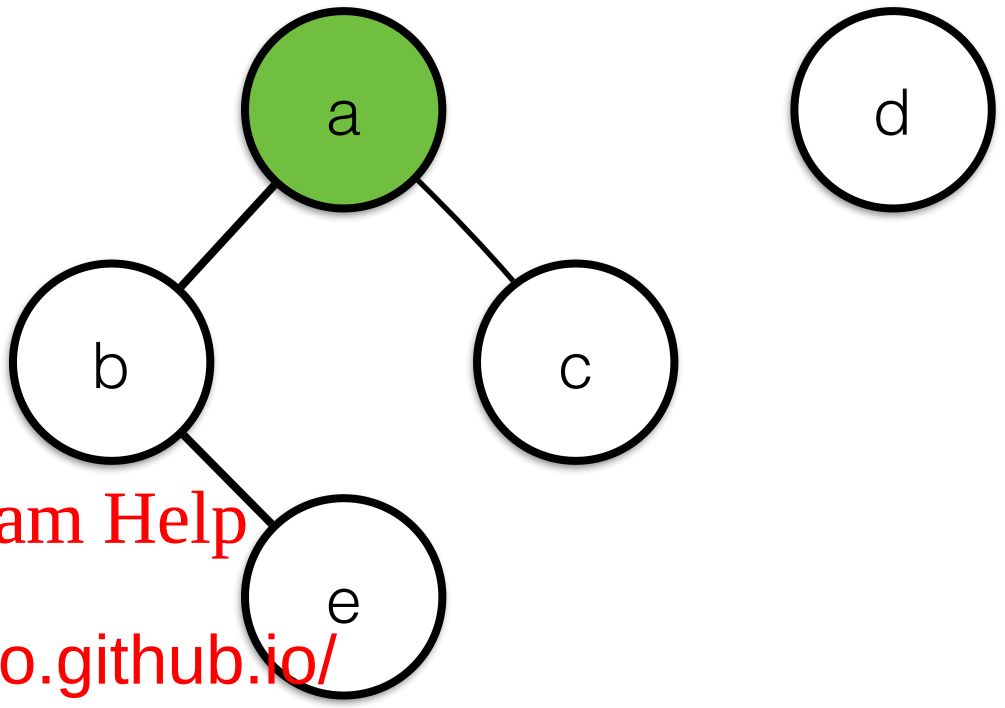
count:
0

DFS(⟨V,E⟩)

Call Stack

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

count:

0

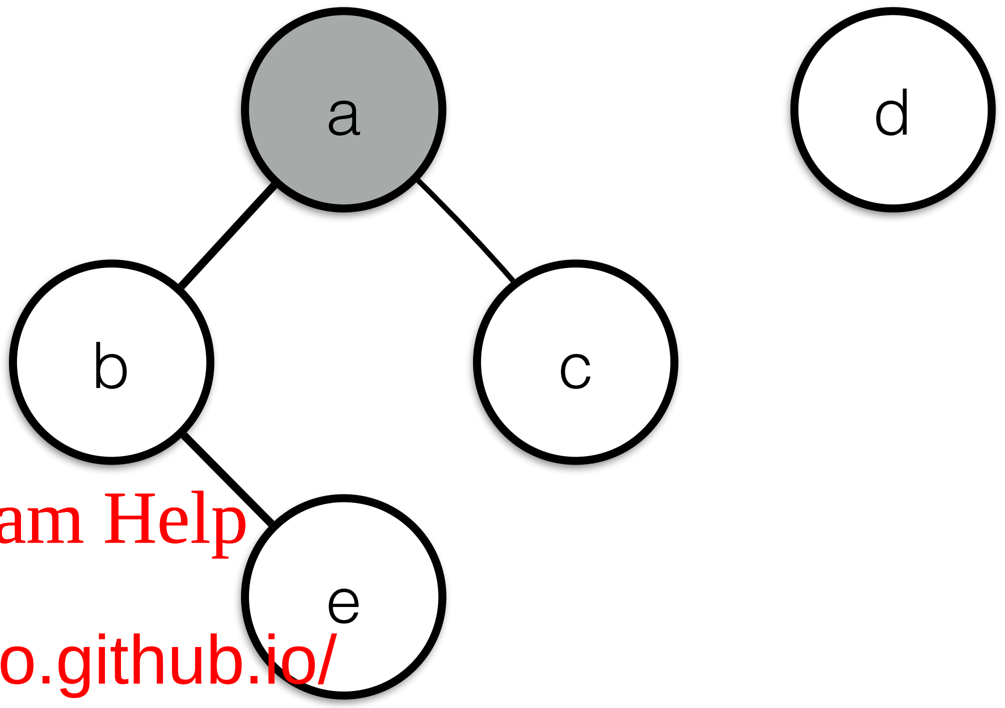
DFSEXPLORE(a)

DFS(⟨V,E⟩)

Call Stack

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

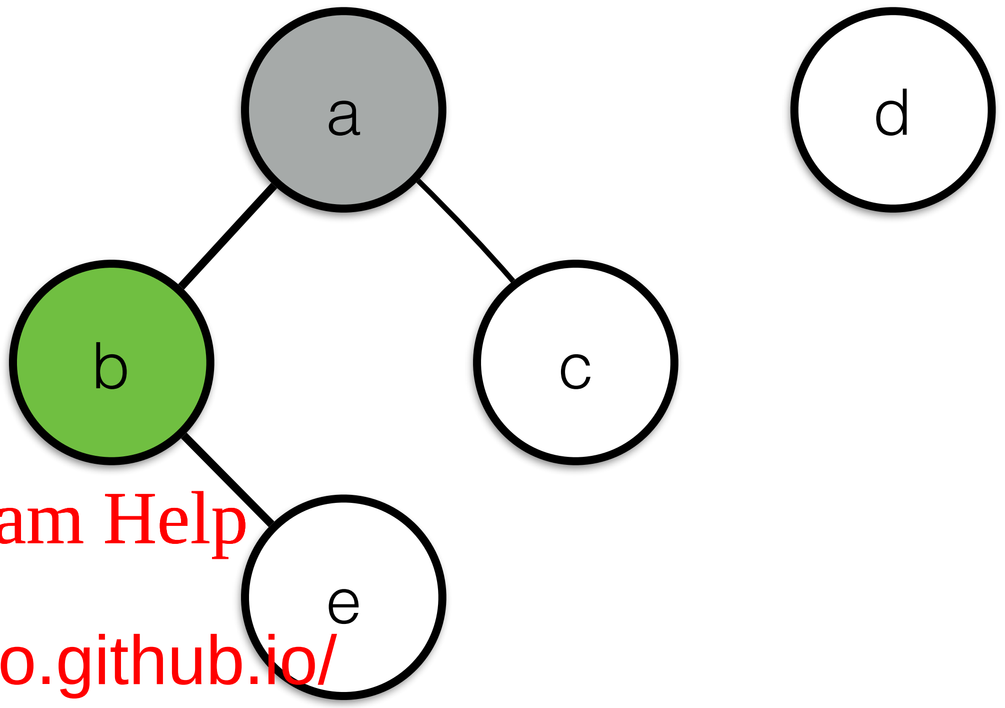
DFSExplore(a)

count: DFS(⟨V,E⟩)

1 **Call Stack**

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

DFSEXPLORE(b)

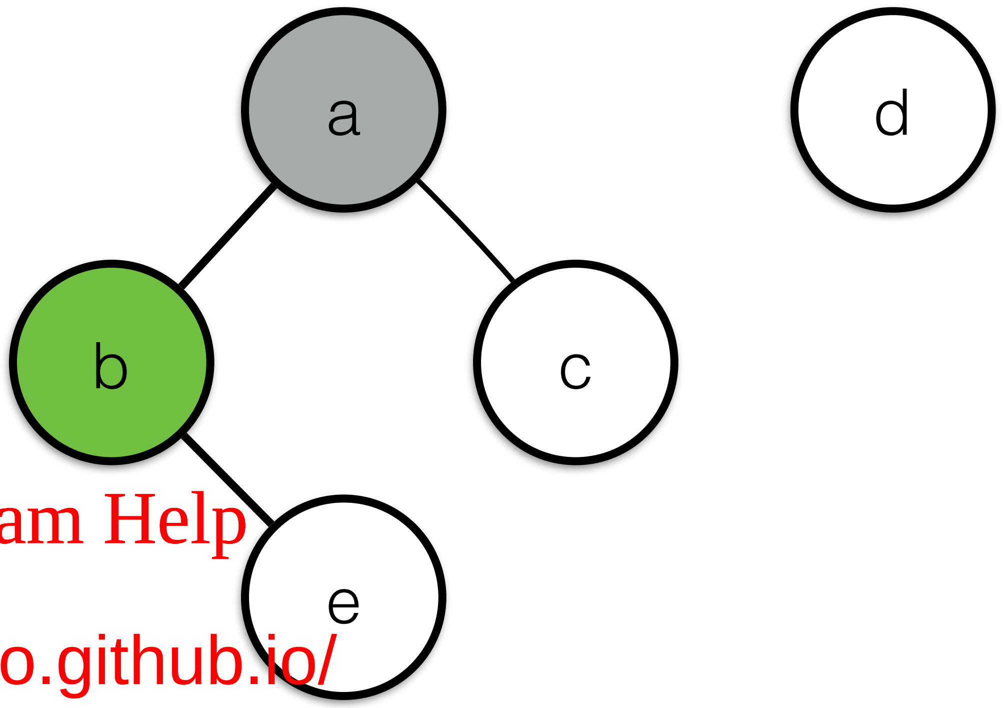
DFSEXPLORE(a)

count: DFS(⟨V,E⟩)

1 **Call Stack**

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



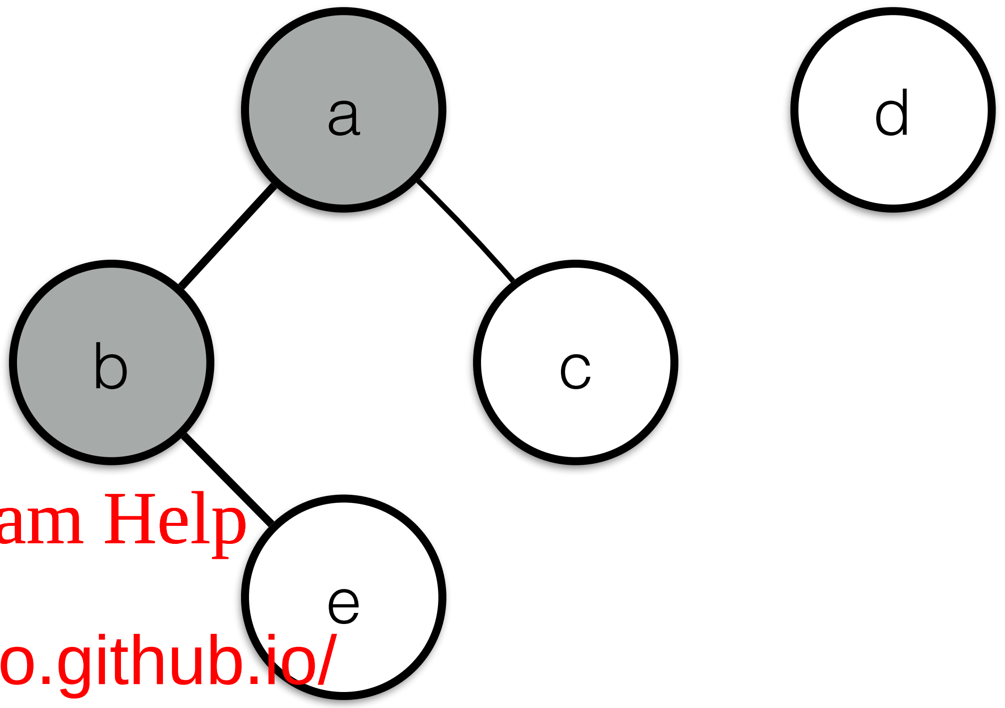
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

DFSEXPLORE(b)
DFSEXPLORE(a)
DFS(⟨V,E⟩)
Call Stack

count:
2

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

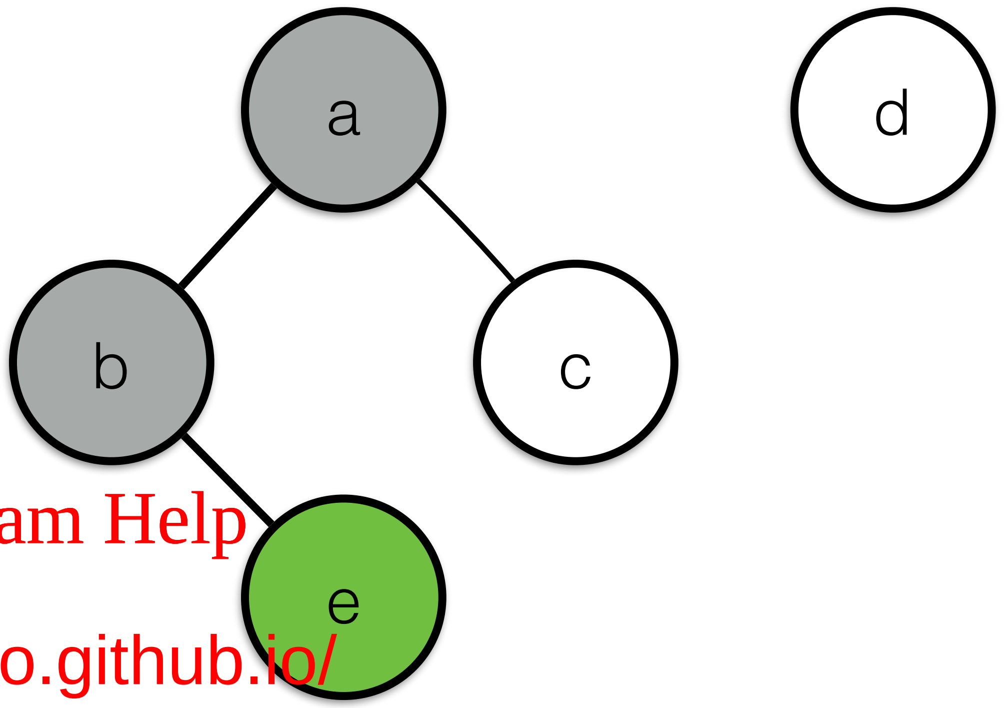
Add WeChat edu_assist_pro

count: 2

DFSEXPLORE(b)
DFSEXPLORE(a)
DFS(⟨V,E⟩)
Call Stack

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

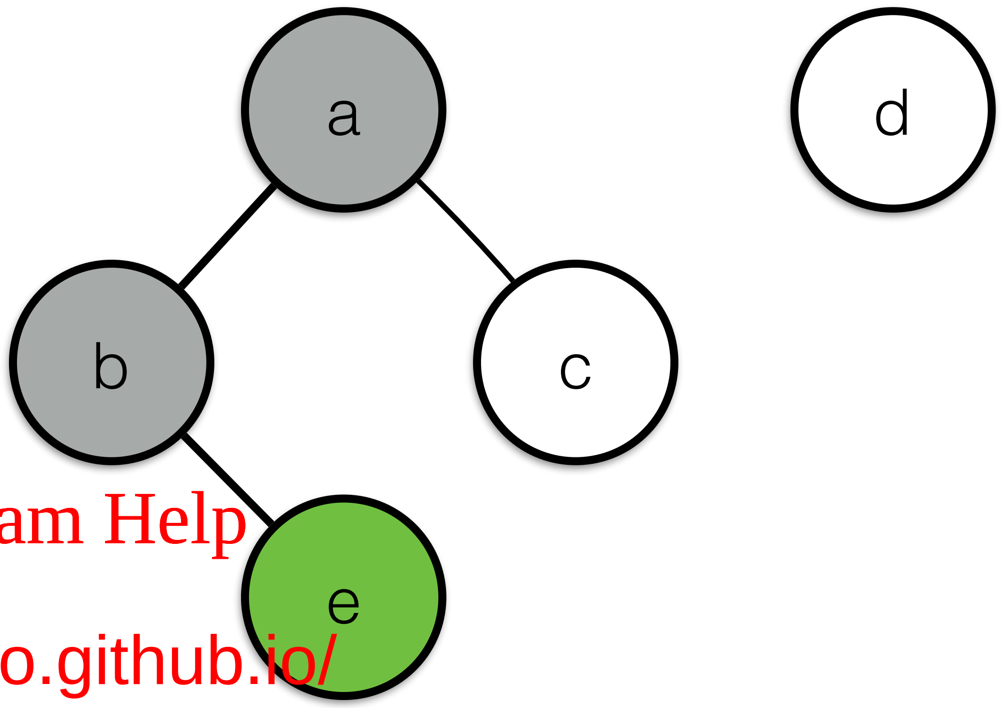
Add WeChat edu_assist_pro

DFSEXPLORE(e)
DFSEXPLORE(b)
DFSEXPLORE(a)
DFS(⟨V,E⟩)
Call Stack

count:
2

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

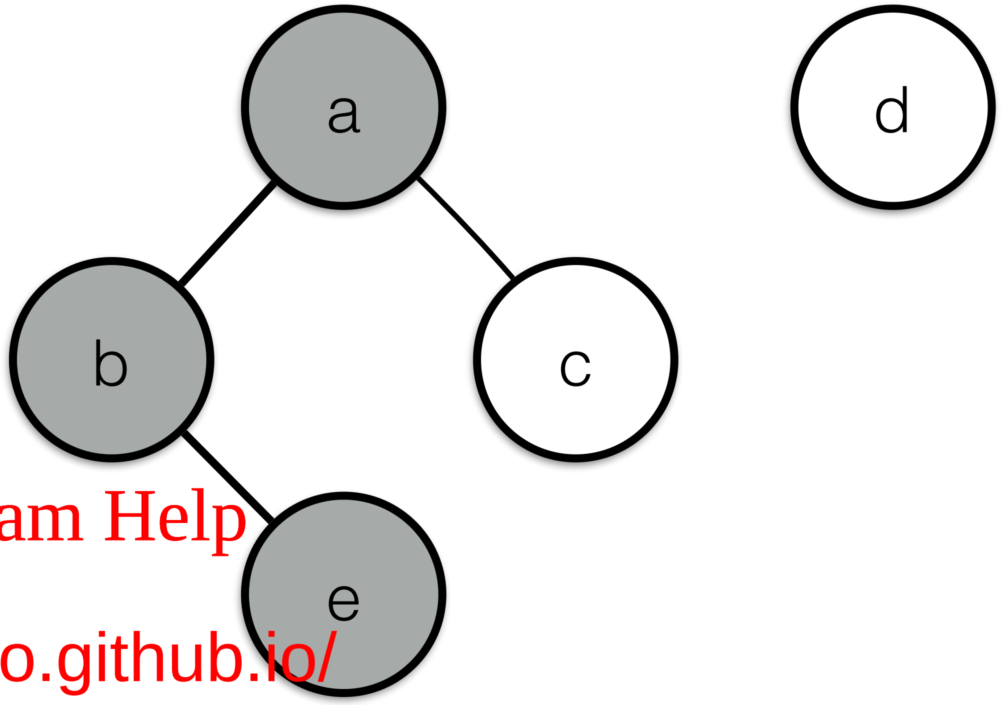
Add WeChat edu_assist_pro

DFSEXPLORE(e)
DFSEXPLORE(b)
DFSEXPLORE(a)
DFS(⟨V,E⟩)
Call Stack

count:
3

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

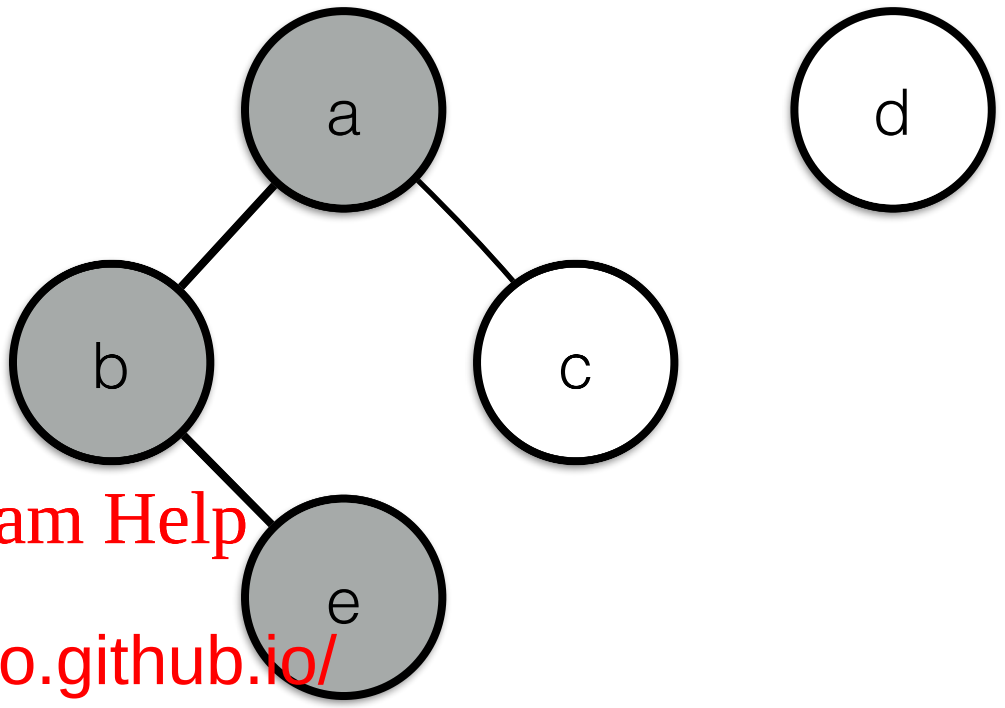
Add WeChat edu_assist_pro

count: 3

DFSEXPLORE(e)
DFSEXPLORE(b)
DFSEXPLORE(a)
DFS(⟨V,E⟩)
Call Stack

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

DFSEXPLORE(b)

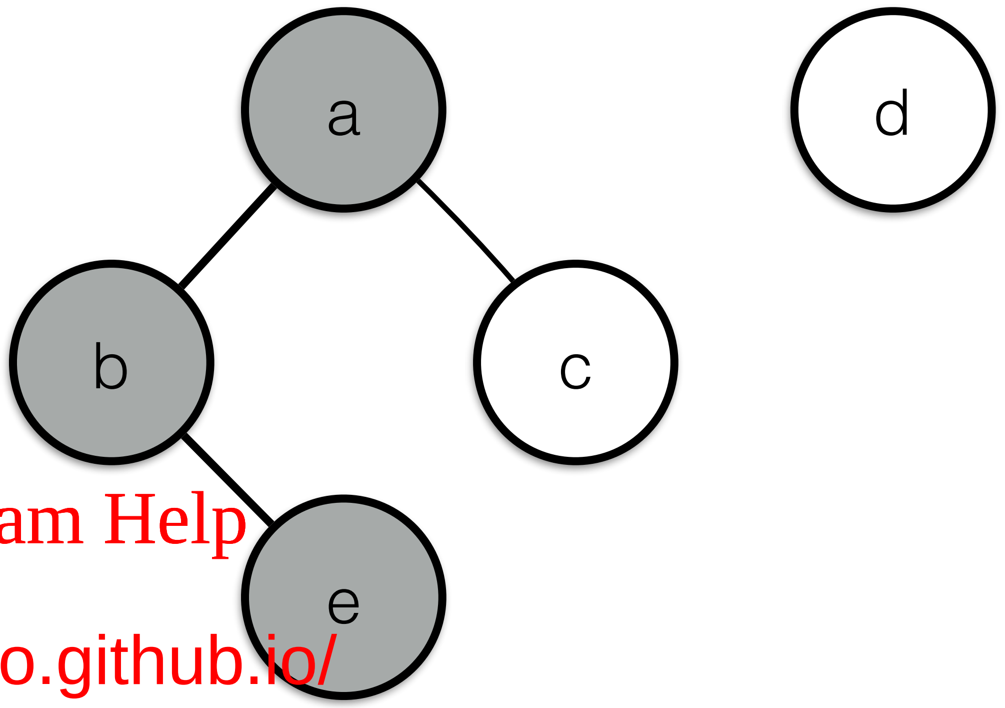
DFSEXPLORE(a)

count: DFS(⟨V,E⟩)

3 **Call Stack**

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

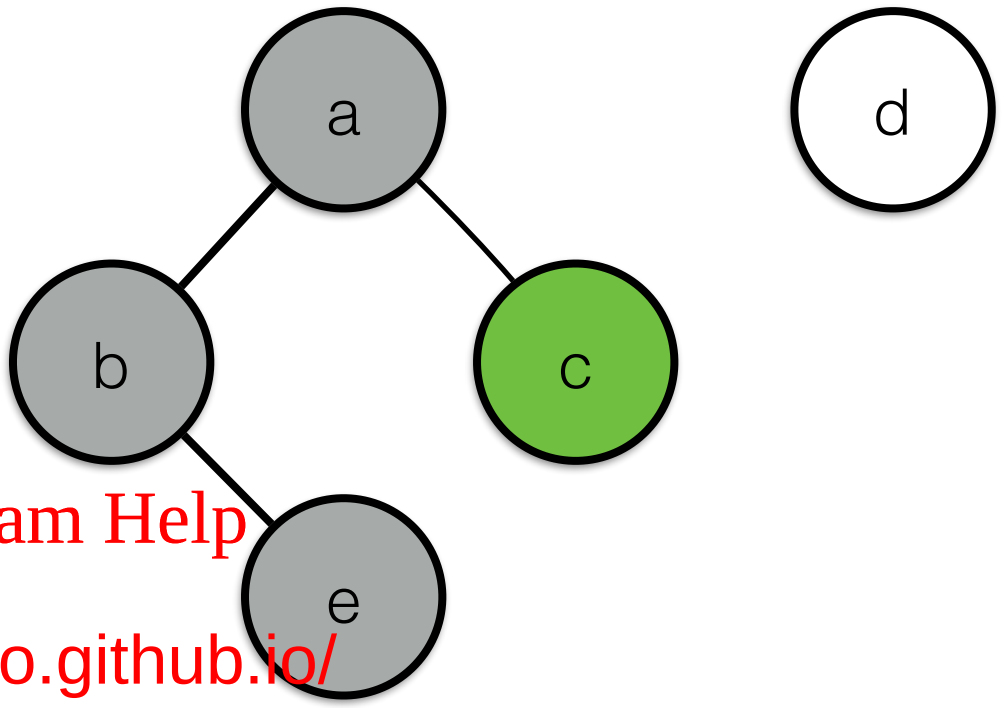
DFSEXPLORE(a)

count: DFS(⟨V,E⟩)

3 **Call Stack**

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

DFSEXPLORE(c)

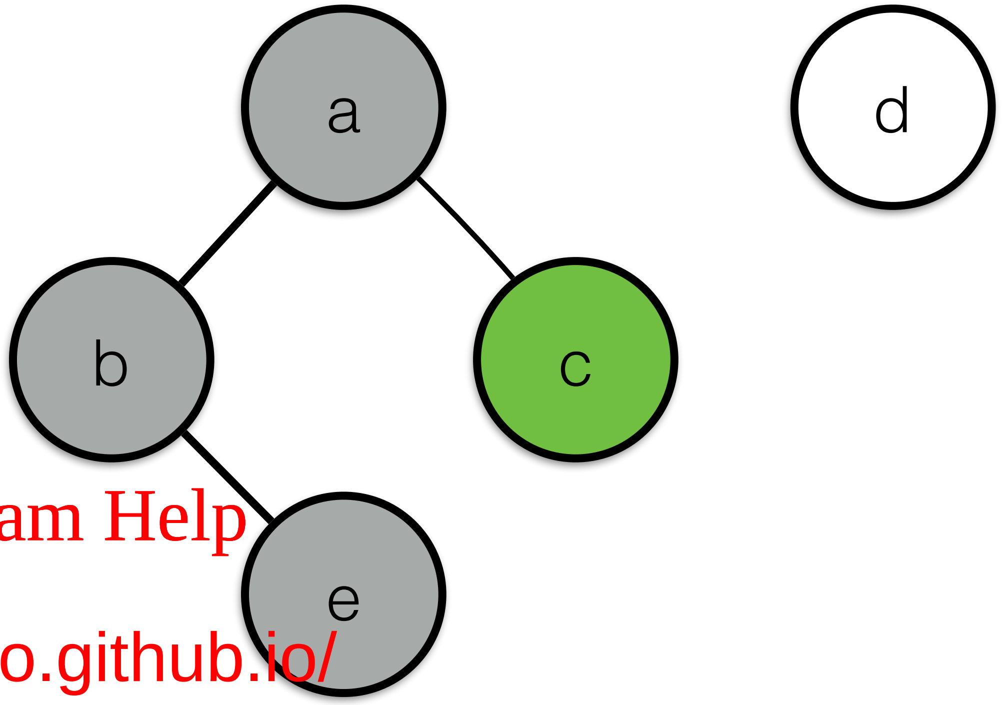
DFSEXPLORE(a)

count: DFS(⟨V,E⟩)

3 **Call Stack**

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



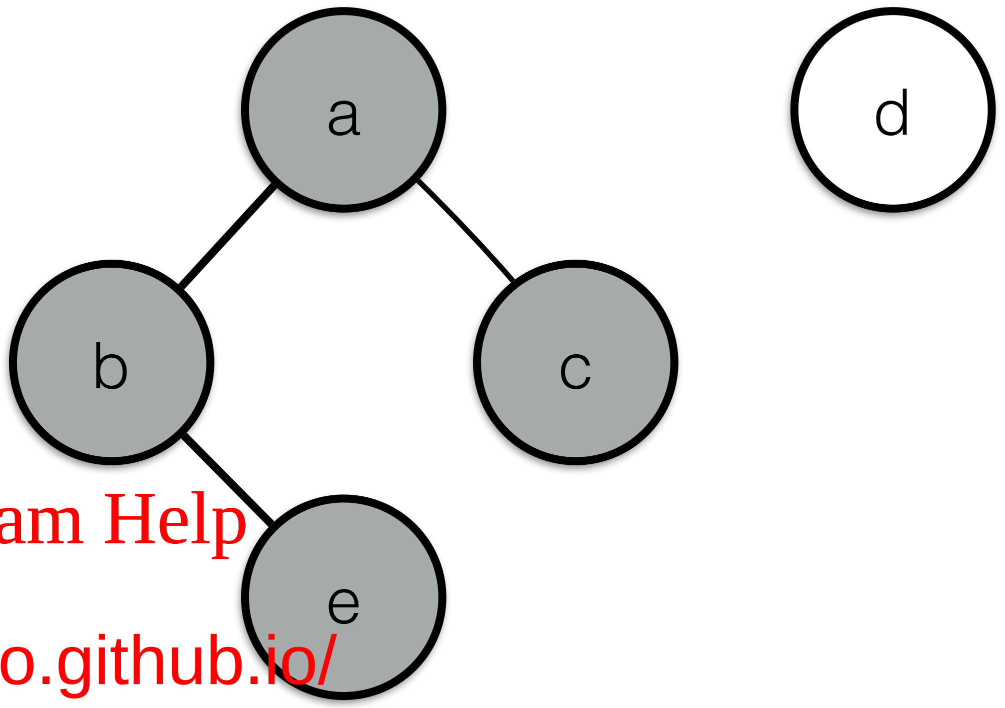
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

count:
4

DFSEXPLORE(c)
DFSEXPLORE(a)
DFS(⟨V,E⟩)
Call Stack

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

DfsExplore(c)

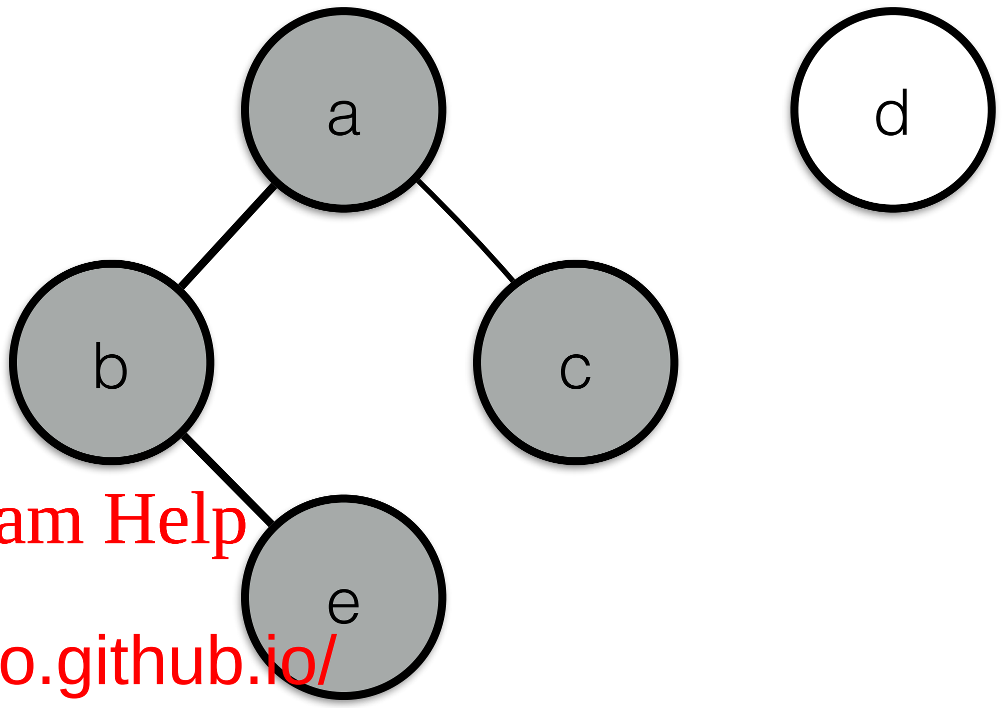
DfsExplore(a)

count: DFS(⟨V,E⟩)

4 **Call Stack**

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

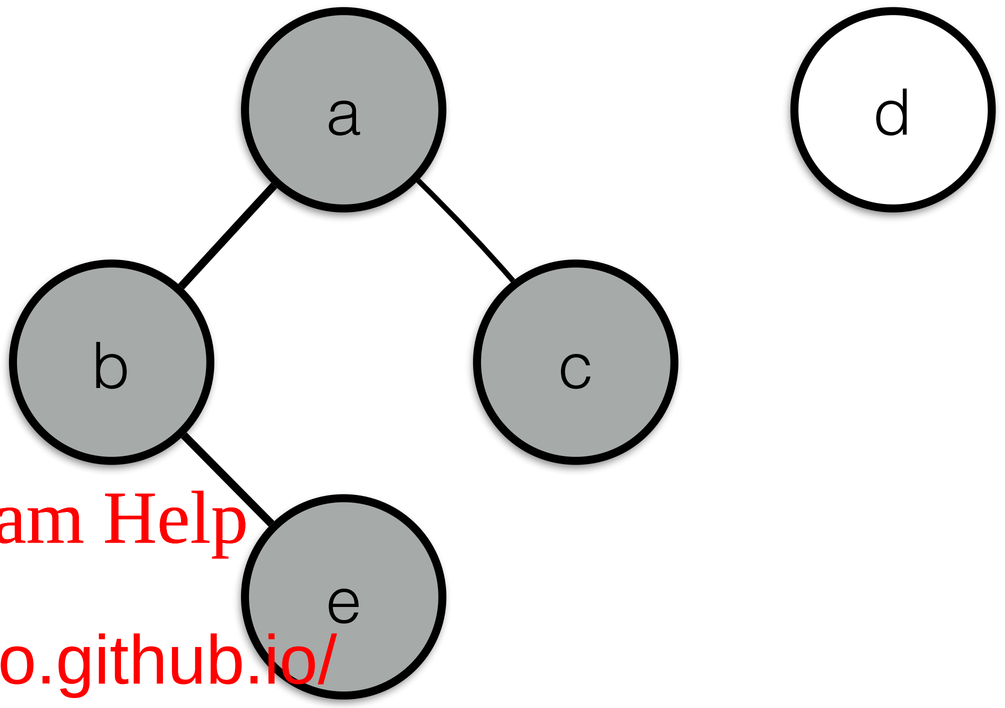
DFSEXPLORE(a)

count: DFS(⟨V,E⟩)

4 **Call Stack**

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

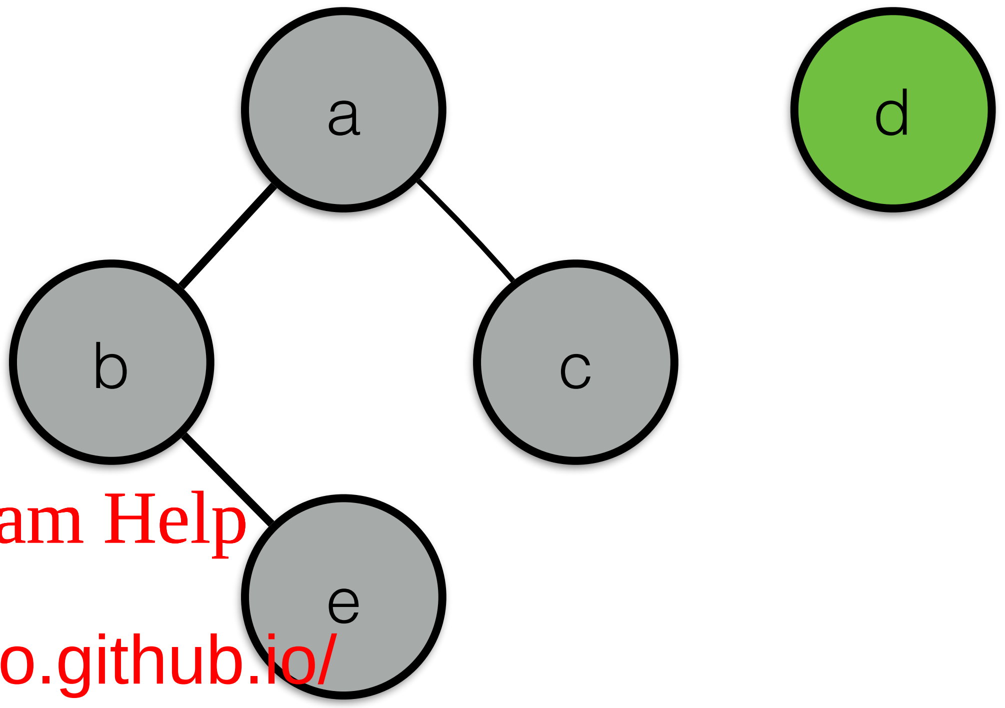
Add WeChat edu_assist_pro

count:
4

DFS(⟨V,E⟩)
Call Stack

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

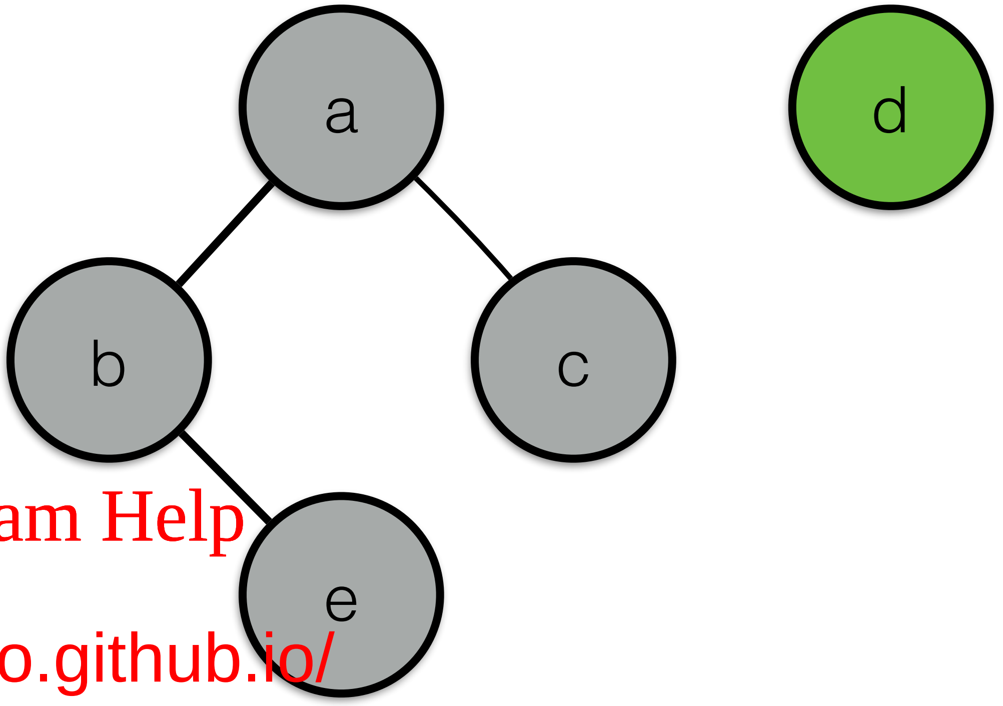
Add WeChat edu_assist_pro

count:
4

DFSEXPLORE(d)
DFS(⟨V,E⟩)
Call Stack

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

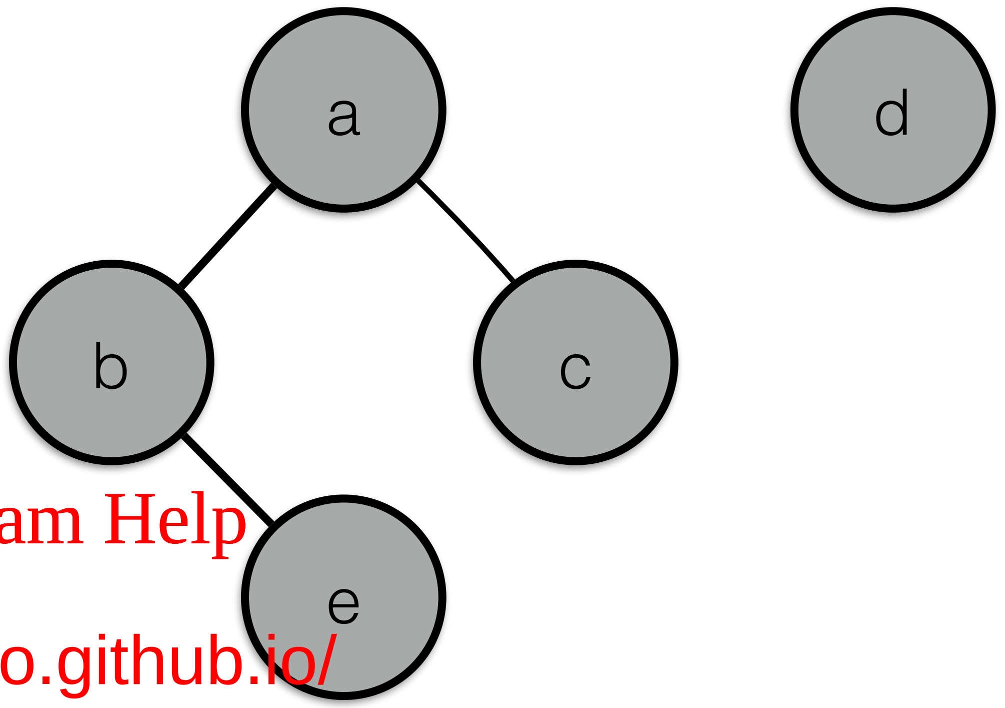
Add WeChat edu_assist_pro

count:
5

DFSEXPLORE(d)
DFS(⟨V,E⟩)
Call Stack

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



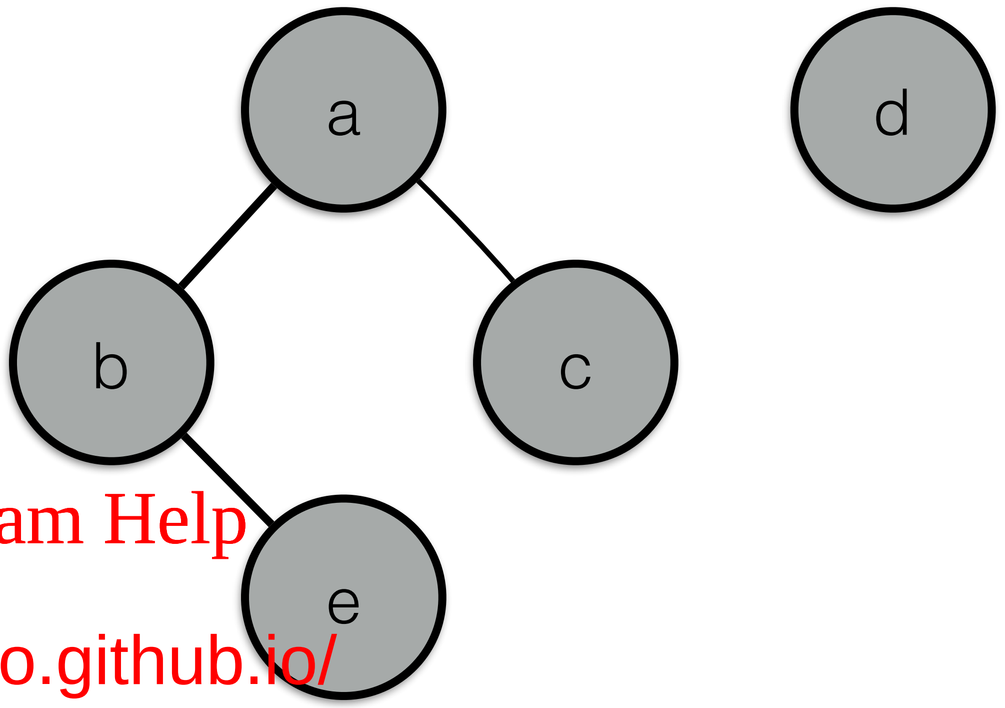
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

count:
5

DFSEXPLORE(d)
DFS(⟨V,E⟩)
Call Stack

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

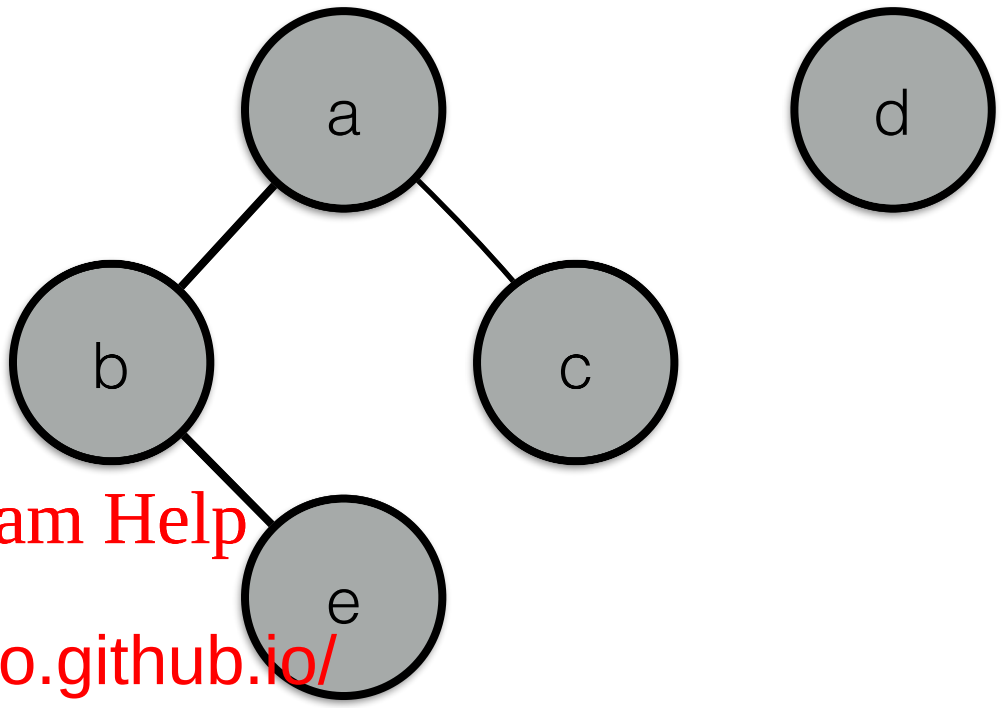
Add WeChat edu_assist_pro

count:
5

DFS(⟨V,E⟩)
Call Stack

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Traversal: Recursive Implementation



Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

count:
5

Call Stack

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Search: Recursive Algorithm Notes

- Works both for directed and undirected graphs.
- The “marking” of nodes is usually done by maintaining a separate array, `mark`, indexed by V .
- For example, when we wrote “mark v with count”, that would be implemented as “`mark`

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

- How to find the nodes adjacent to on the graph representation used.
- Using an adjacency **matrix** `adj`, we need to consider `adj[v,w]` for each w in V . Here the complexity of graph traversal is Θ($|V|^2$).
- Using adjacency **lists**, for each v , we traverse the list `adj[v]`. In this case, the complexity of traversal is Θ(|V| + |E|).

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Applications of Depth-First Search (DFS)

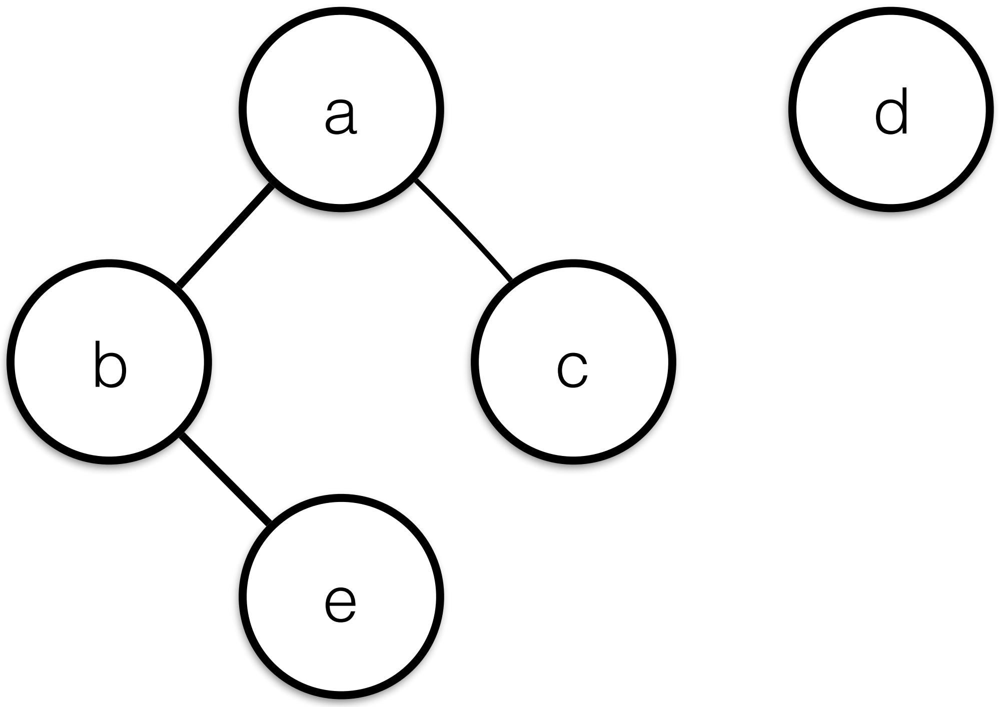
- Easy to adapt DFS to decide if a graph is connected.

- How?

Assignment Project Exam Help

https://eduassistpro.github.io/

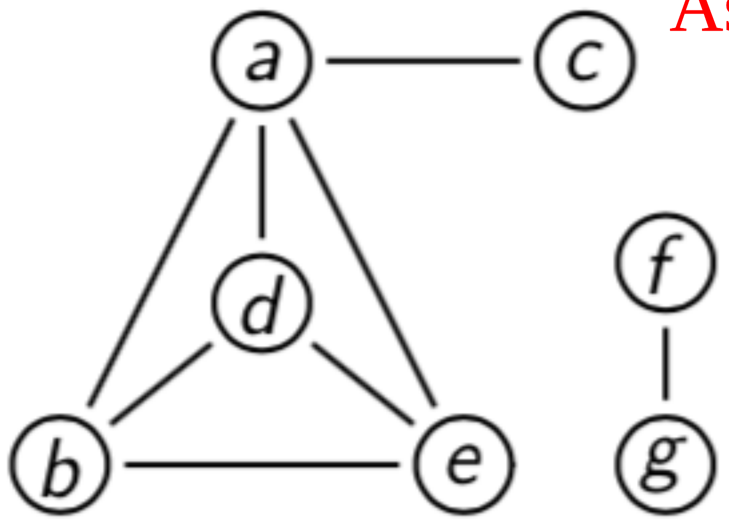
Add WeChat edu_assist_pro



Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Search: Node Orderings

- We can order nodes **either** by the order in which they get **pushed onto** the stack, or by the order in which they are **popped from** the stack



Levitin's compact stack notation

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

The first subscripts give the order in which nodes are pushed, the second the order in which they are popped off the stack.

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First Search Forest

DFS can be depicted by the resulting **DFS Forest** (DFS **Tree** for a connected graph)

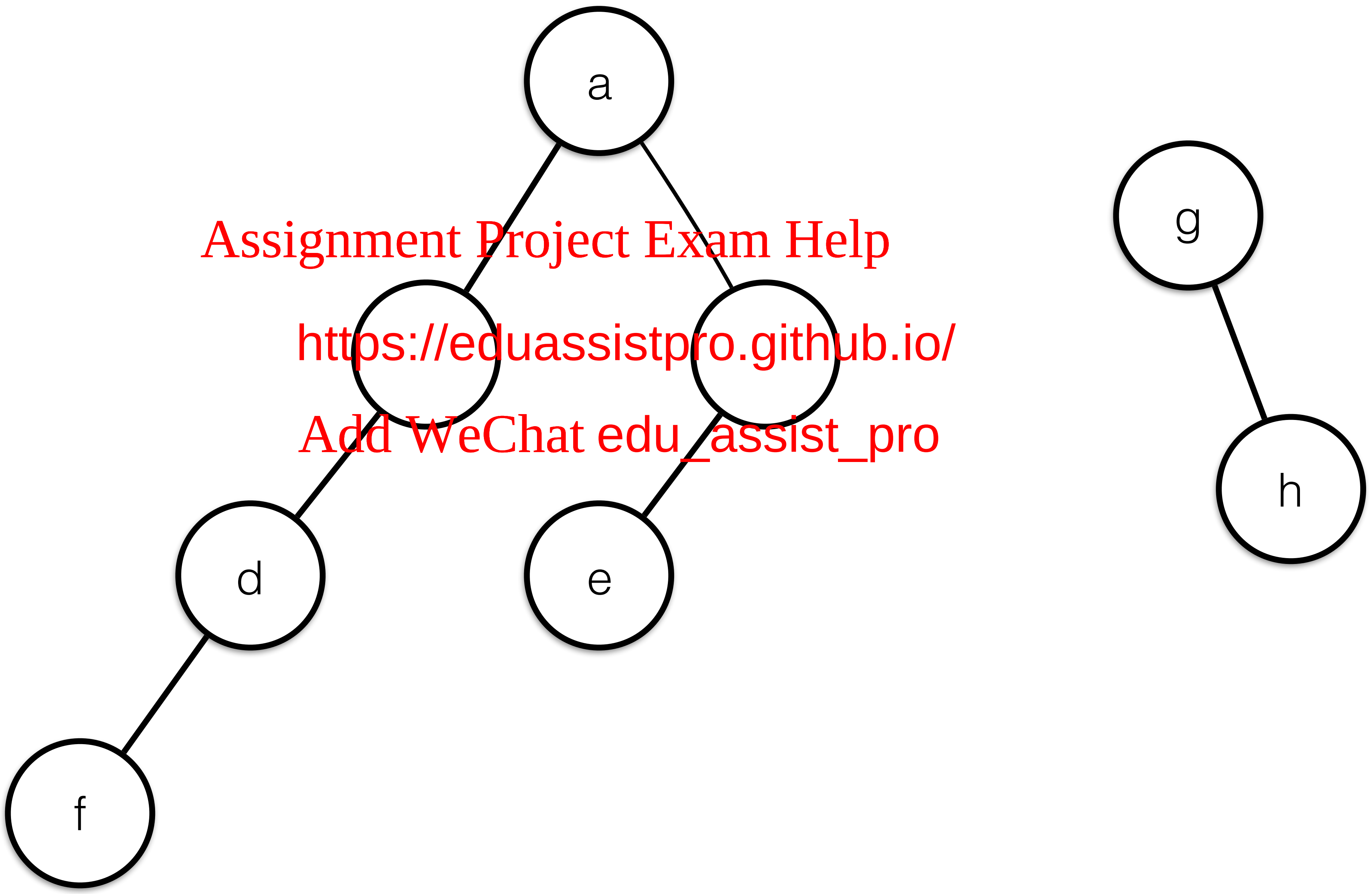
Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Traversal



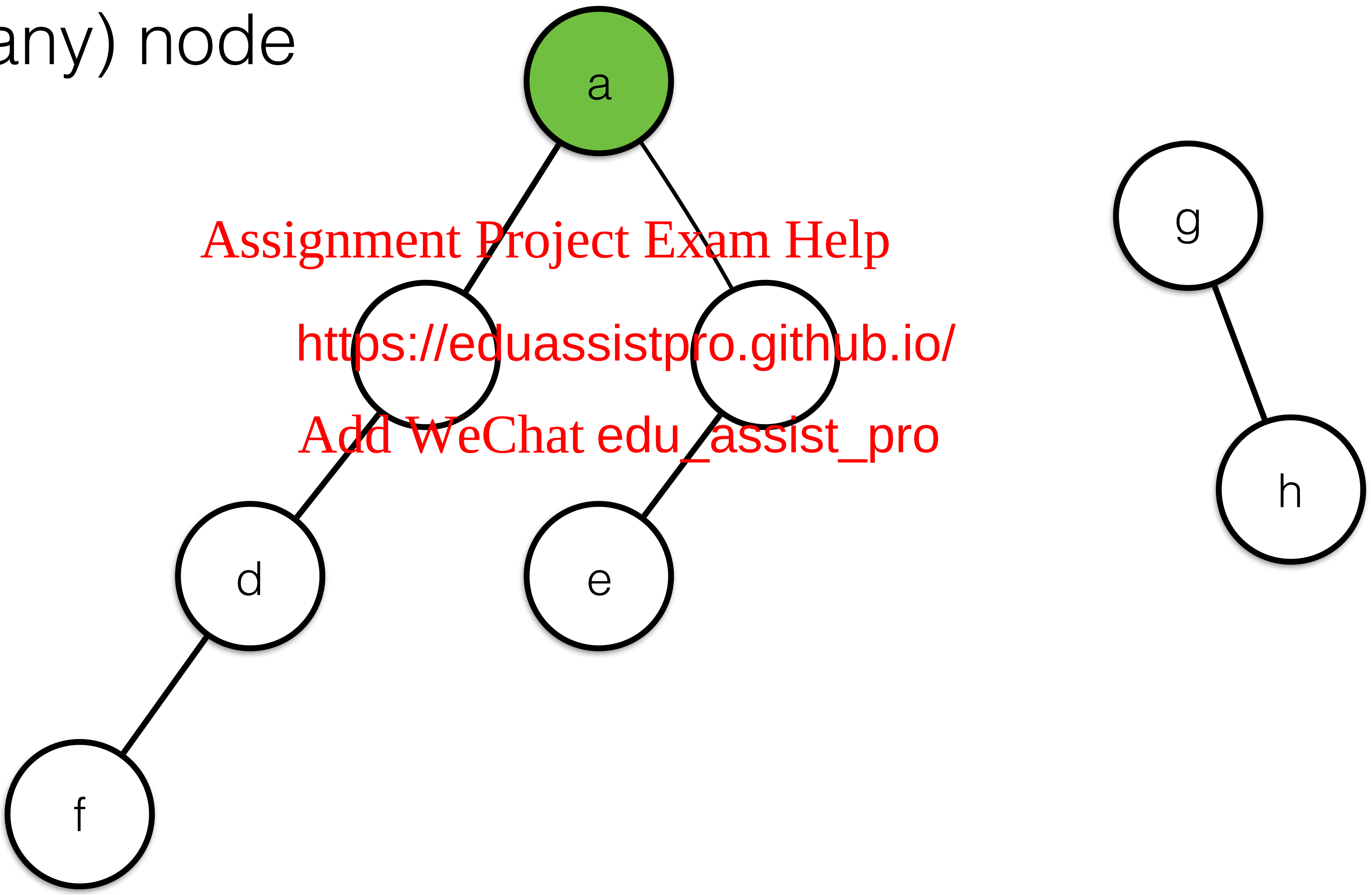
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Traversal

Nodes visited in this order: a

Start with (any) node



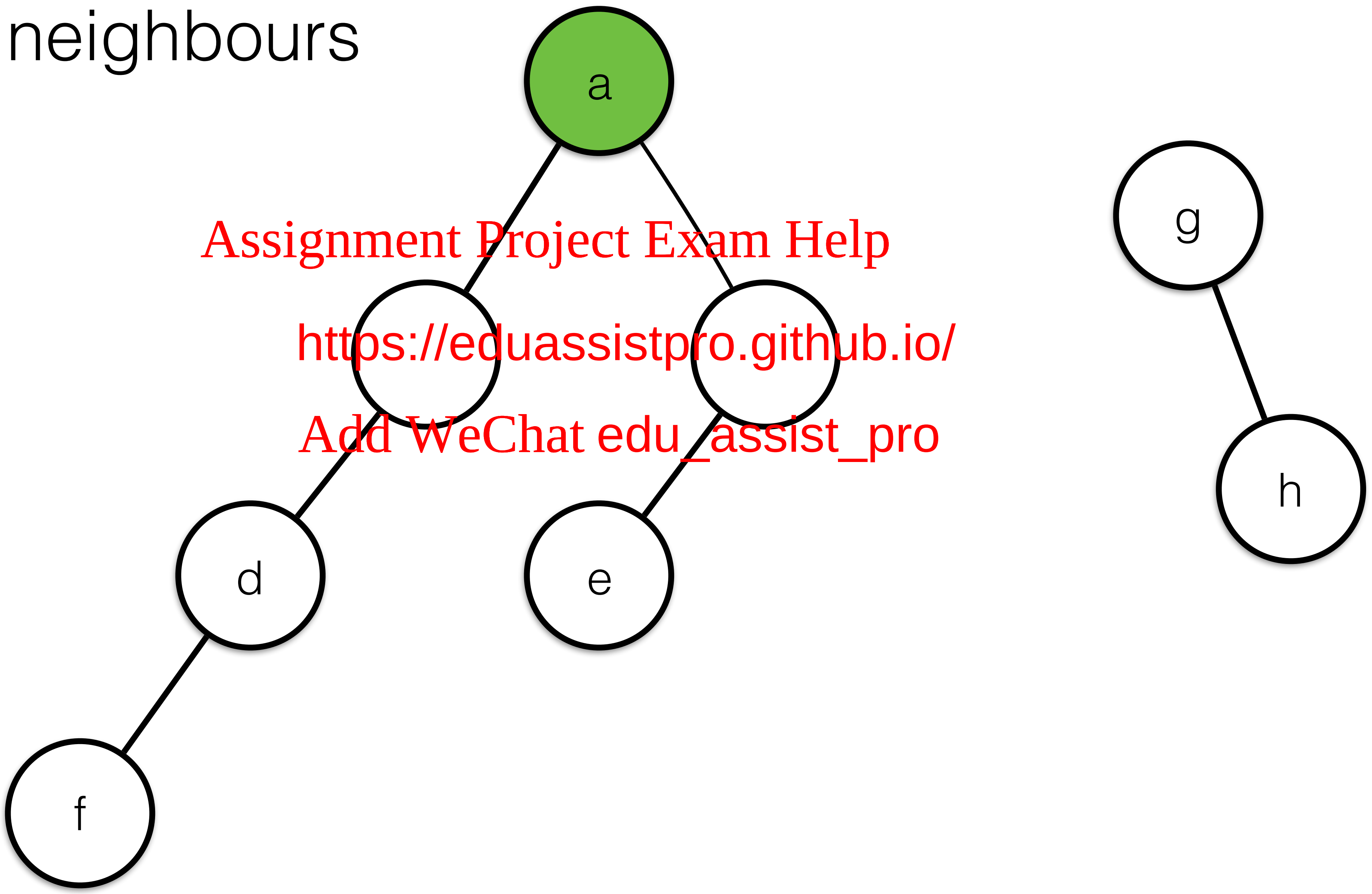
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Traversal

Nodes visited in this order: a

Visit all of its neighbours



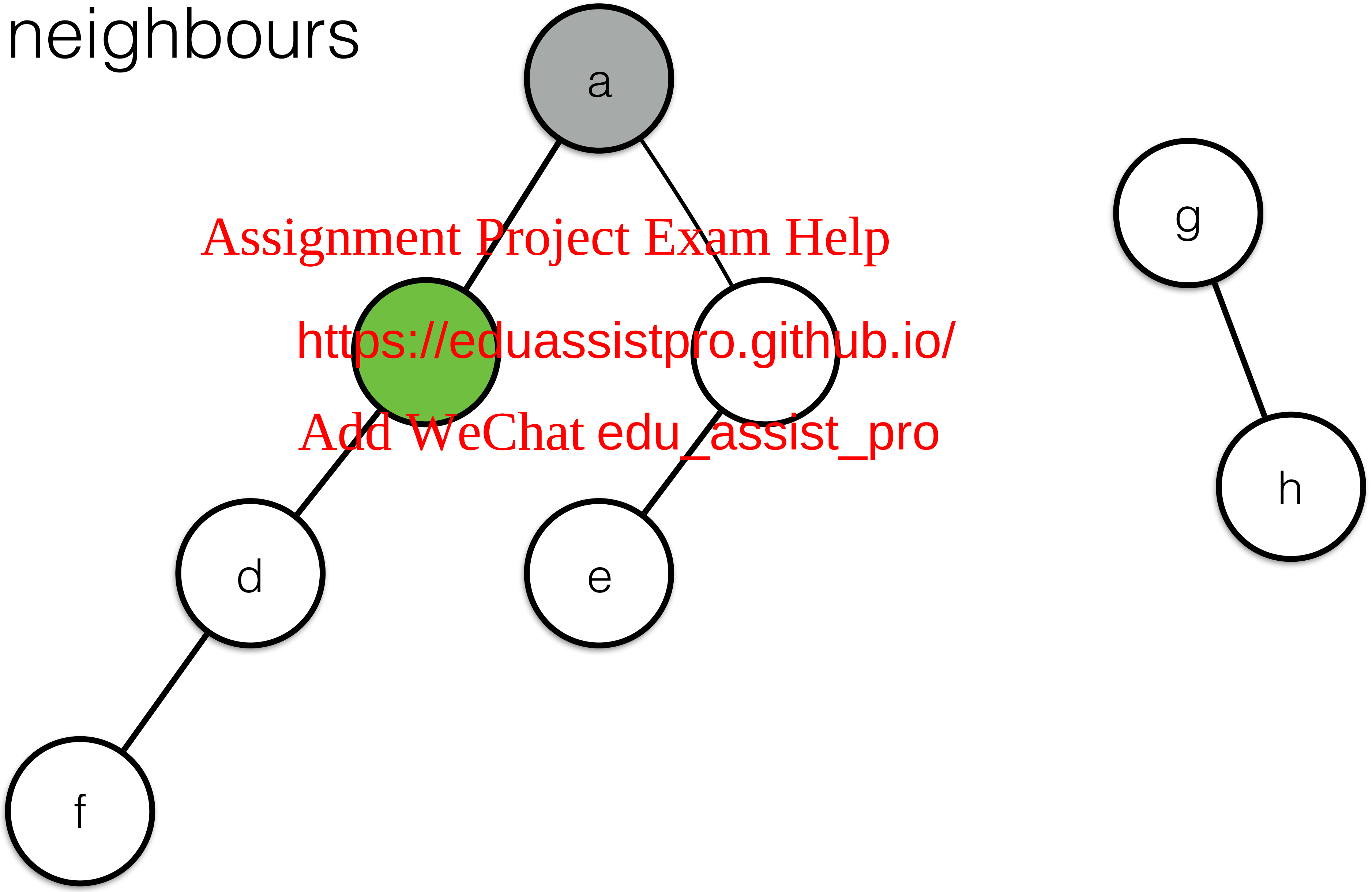
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Traversal

Nodes visited in this order: a b

Visit all of its neighbours



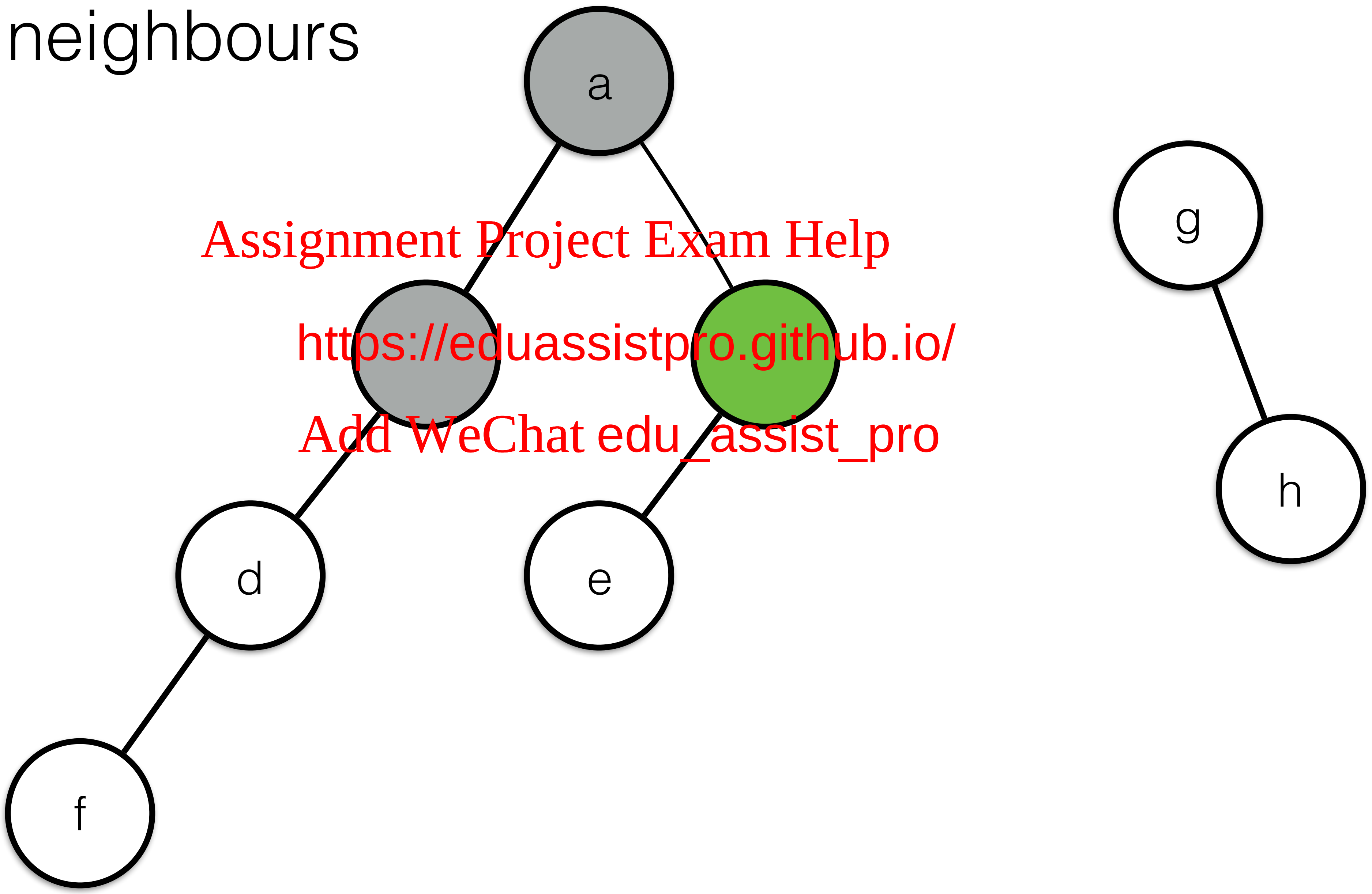
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Traversal

Nodes visited in this order: a b c

Visit all of its neighbours



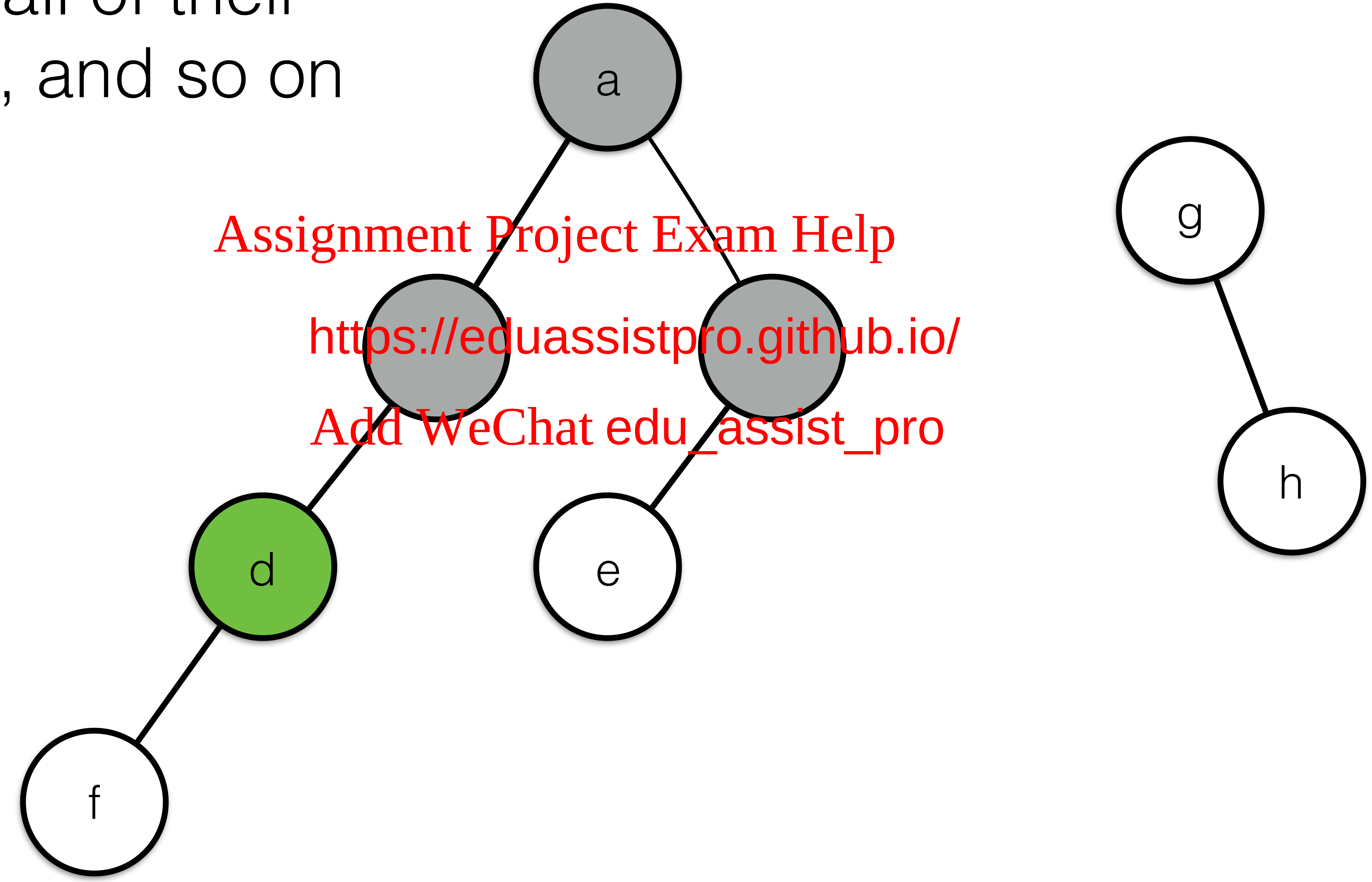
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Traversal

Nodes visited in this order: a b c d

Then visit all of their neighbours, and so on



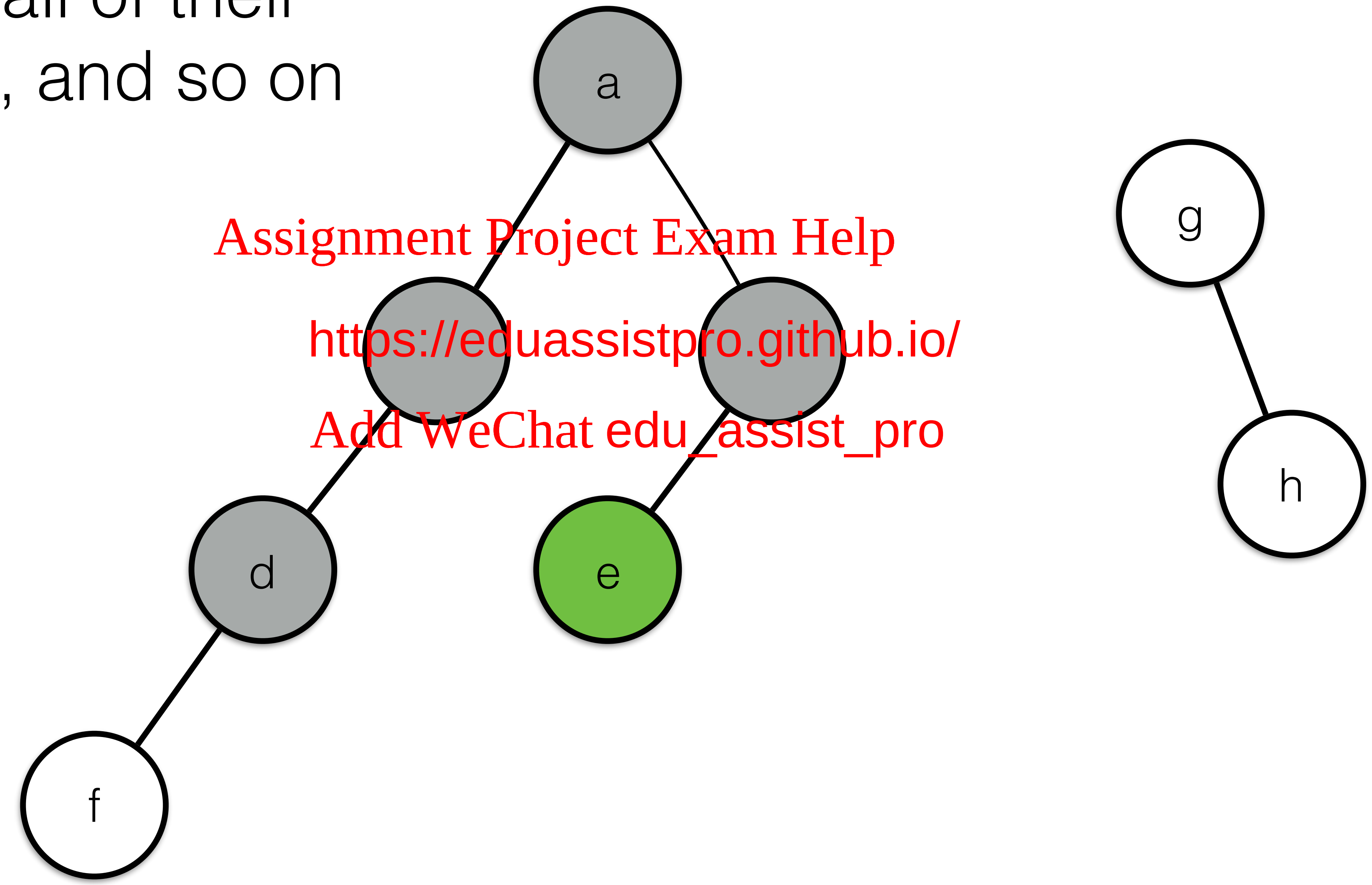
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Traversal

Nodes visited in this order: a b c d e

Then visit all of their neighbours, and so on



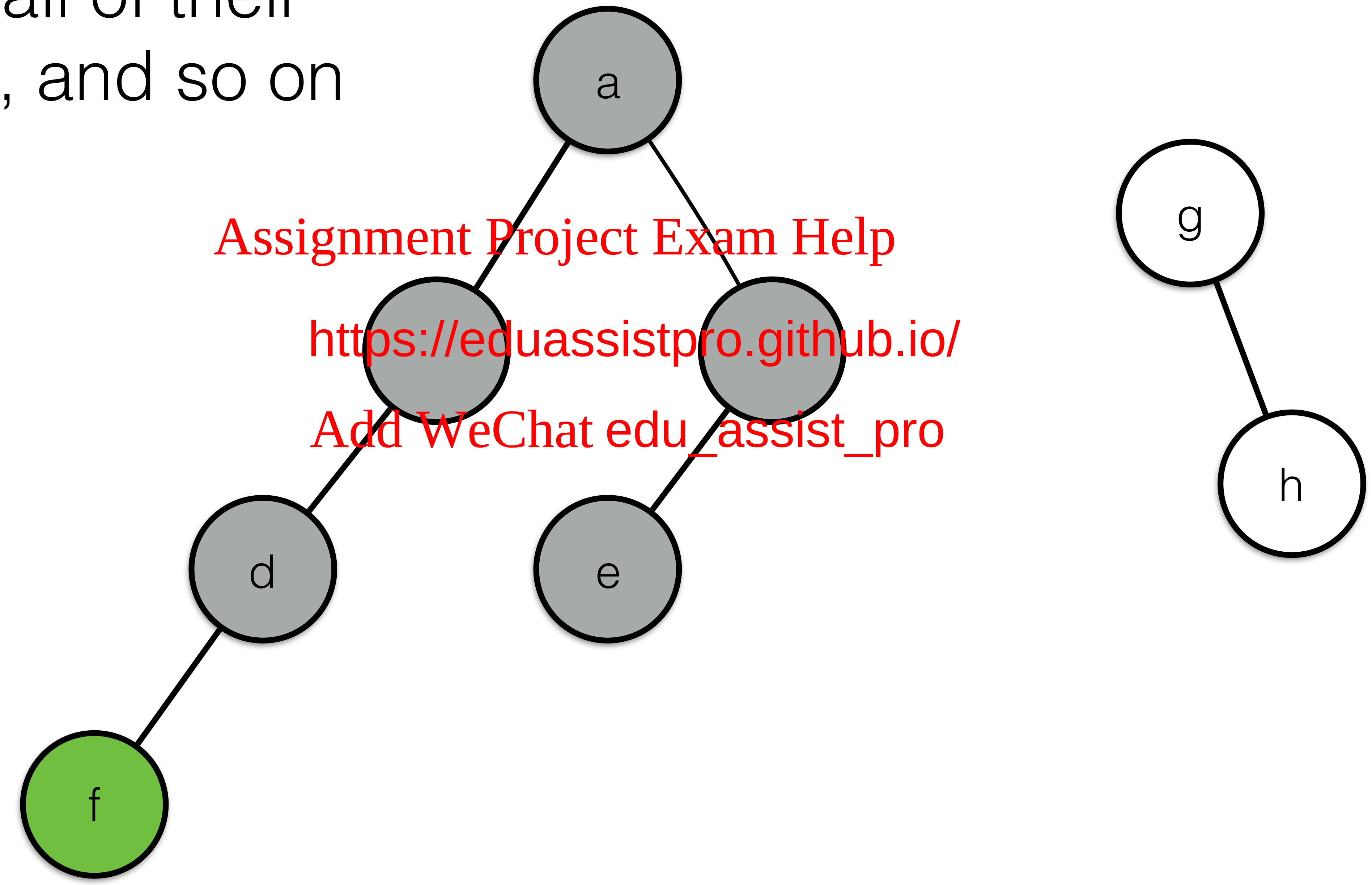
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Traversal

Nodes visited in this order: a b c d e f

Then visit all of their neighbours, and so on



Assignment Project Exam Help

https://eduassistpro.github.io/

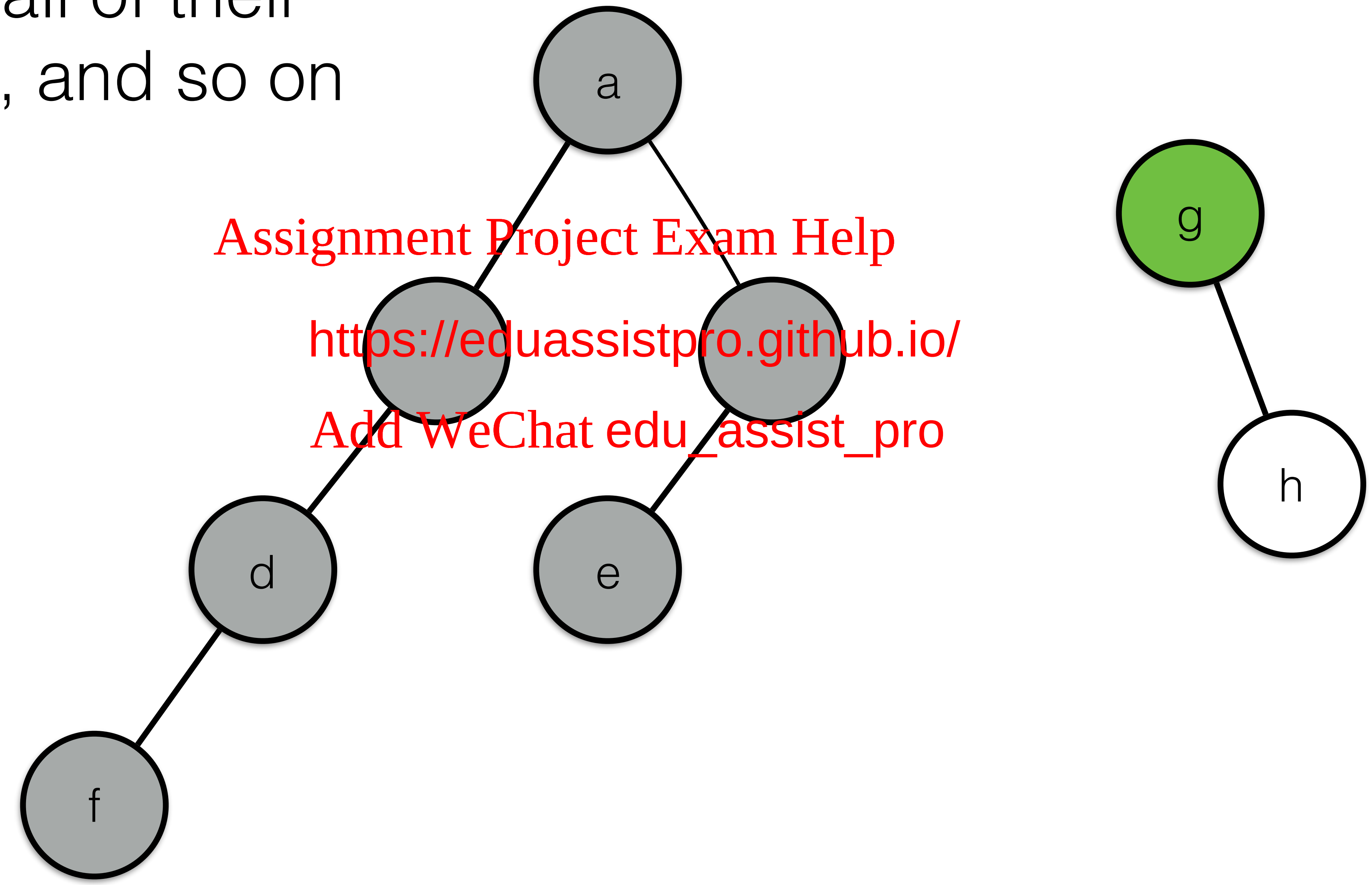
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Traversal

Nodes visited in this order: a b c d e f g

Then visit all of their neighbours, and so on



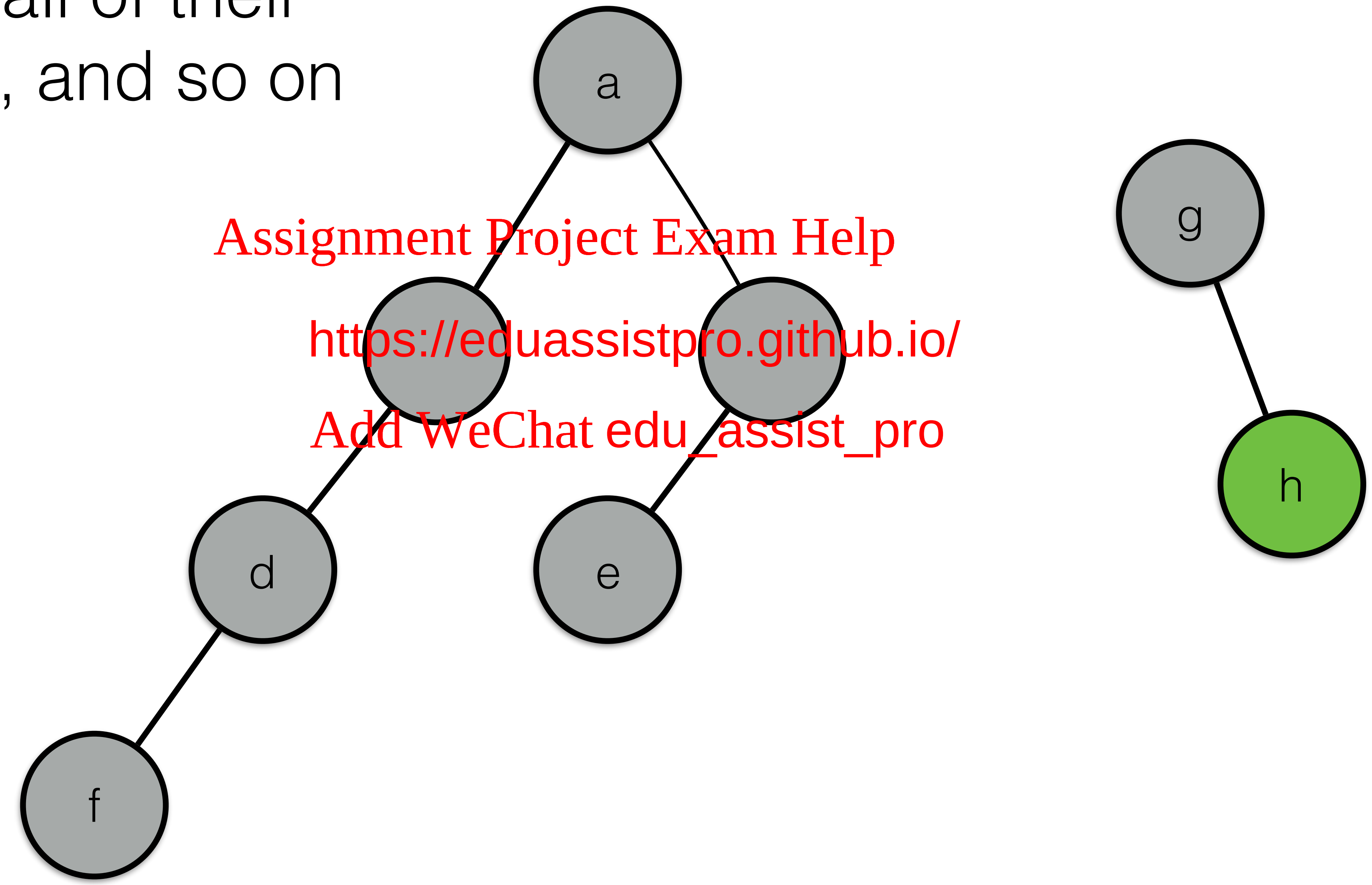
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Traversal

Nodes visited in this order: a b c d e f g h

Then visit all of their neighbours, and so on



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First vs Breadth-First Search

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Typical Depth-First Search

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Depth-First vs Breadth-First Search

Assignment Project Exam Help

https://eduassistpro.github.io/

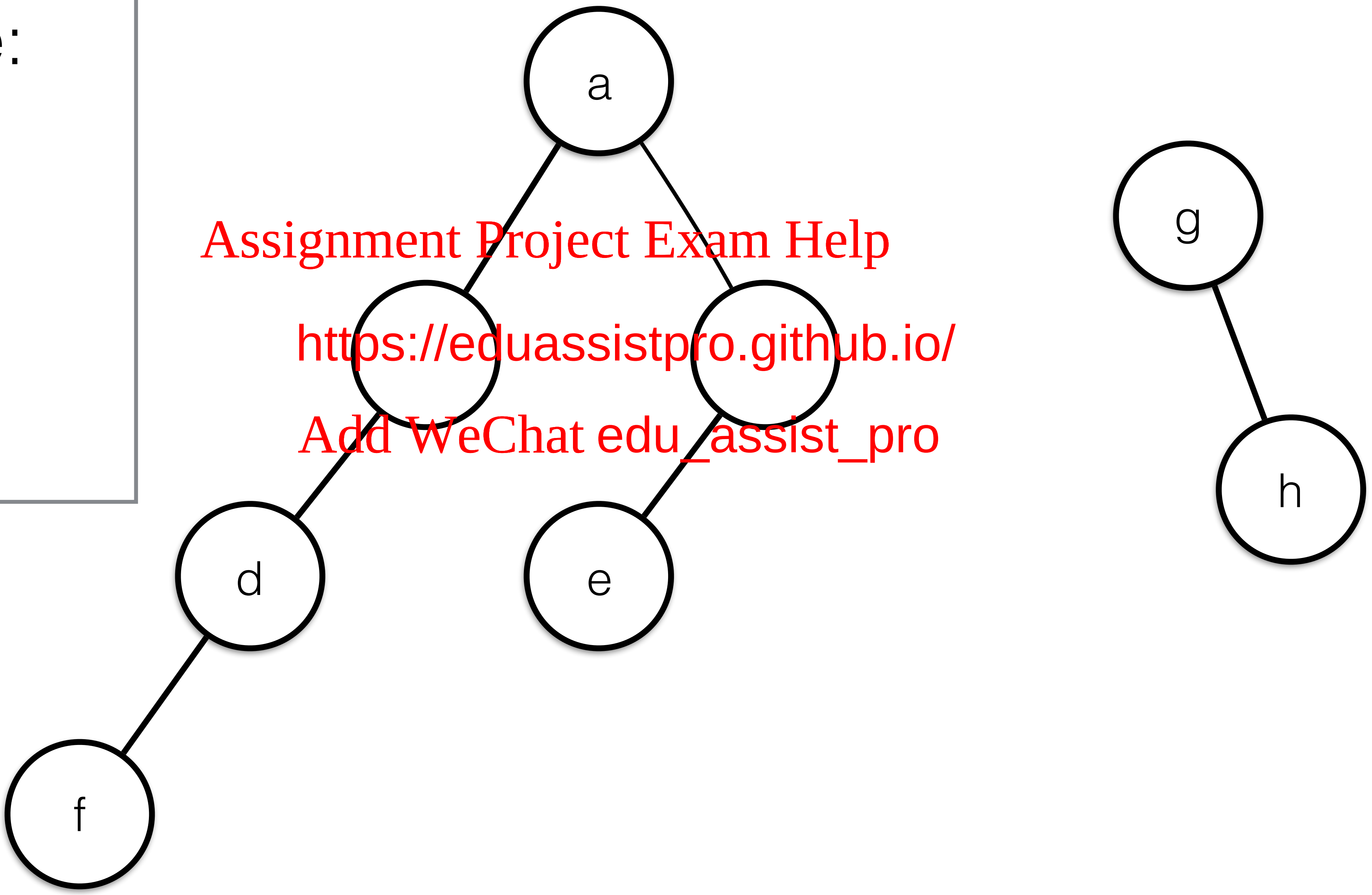
Add WeChat edu_assist_pro

Typical Breadth-First Search

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal Queue:

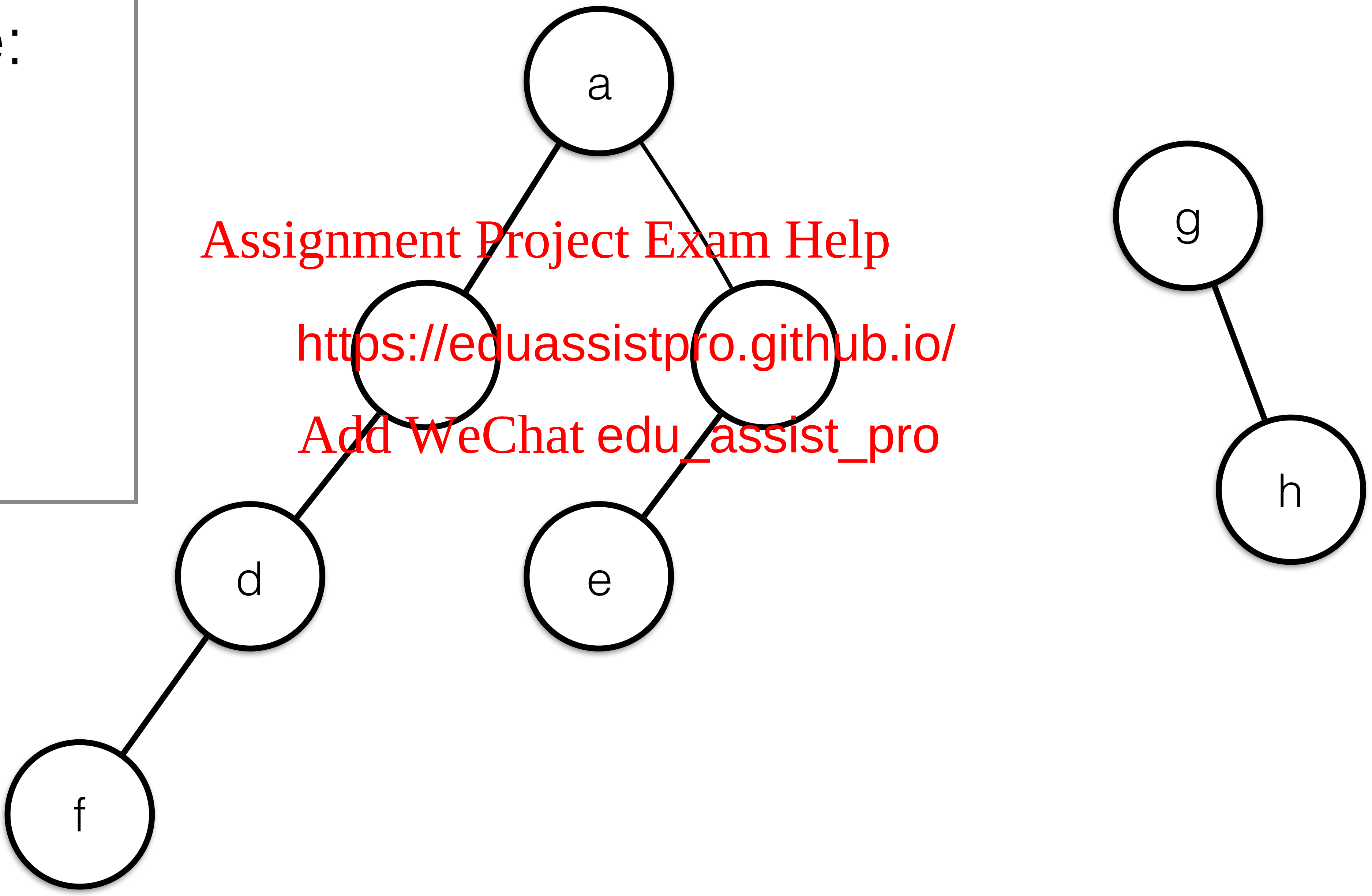


Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal
Queue:
a

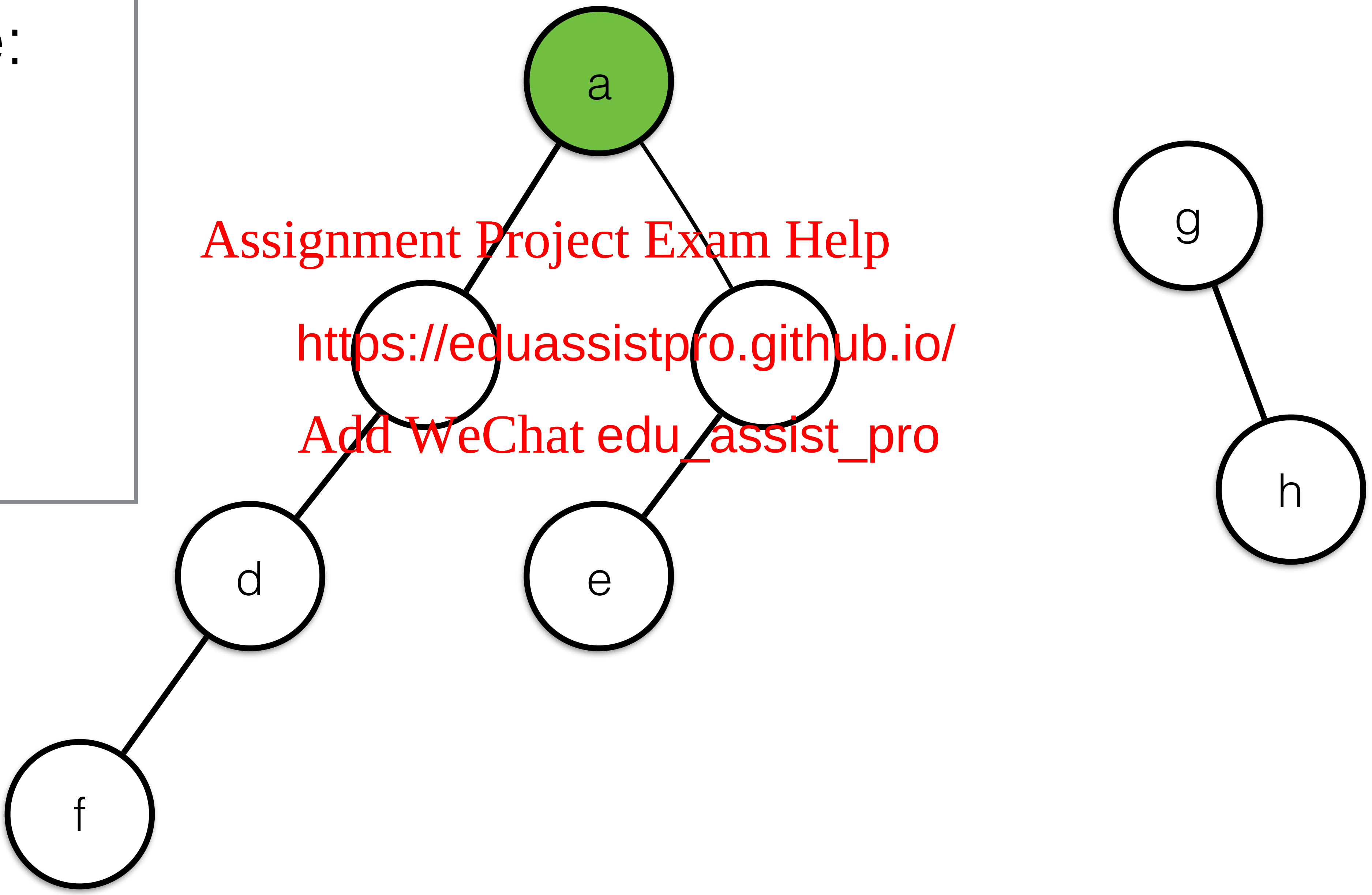


Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal Queue:



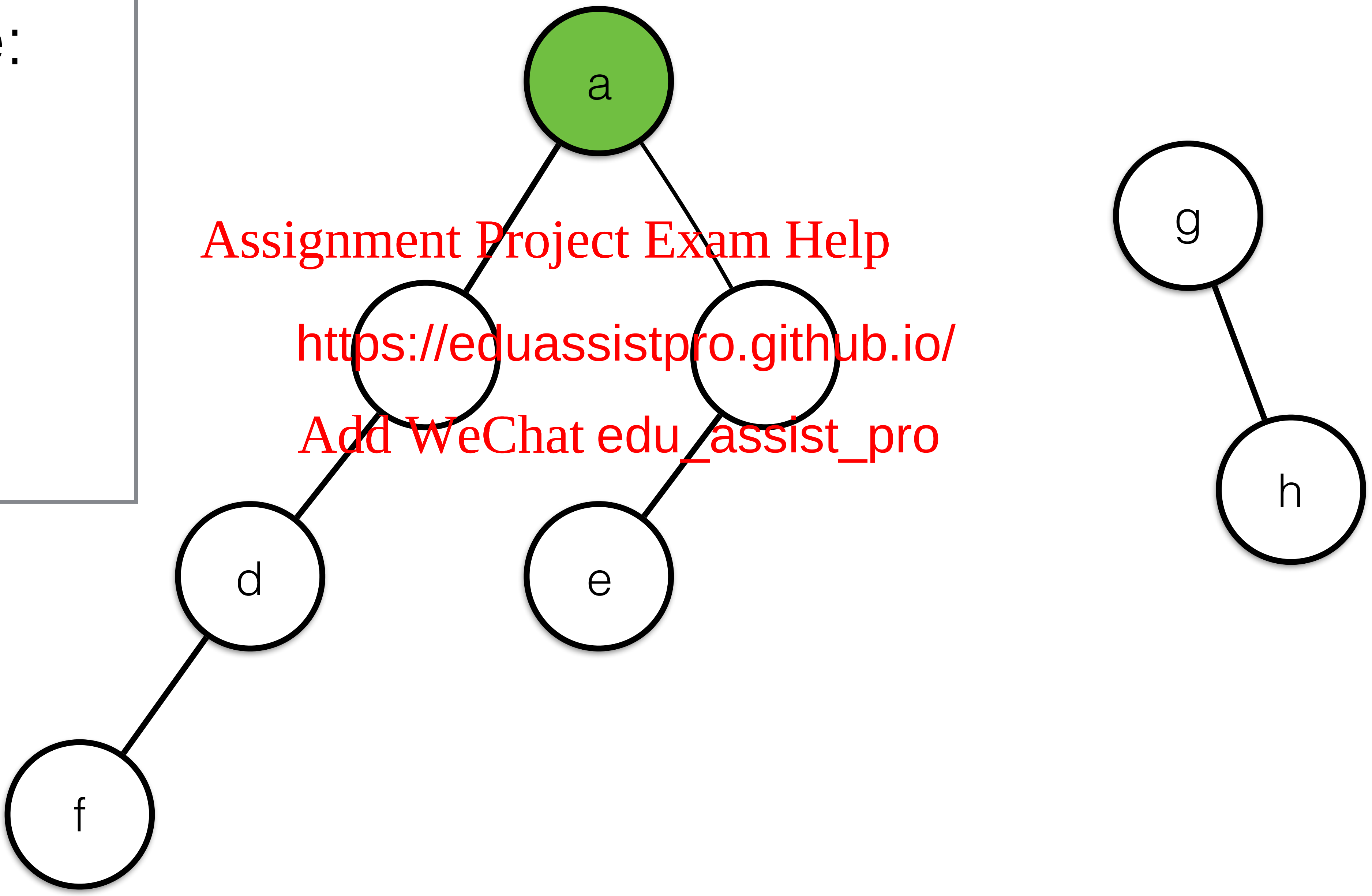
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal Queue:

b
c

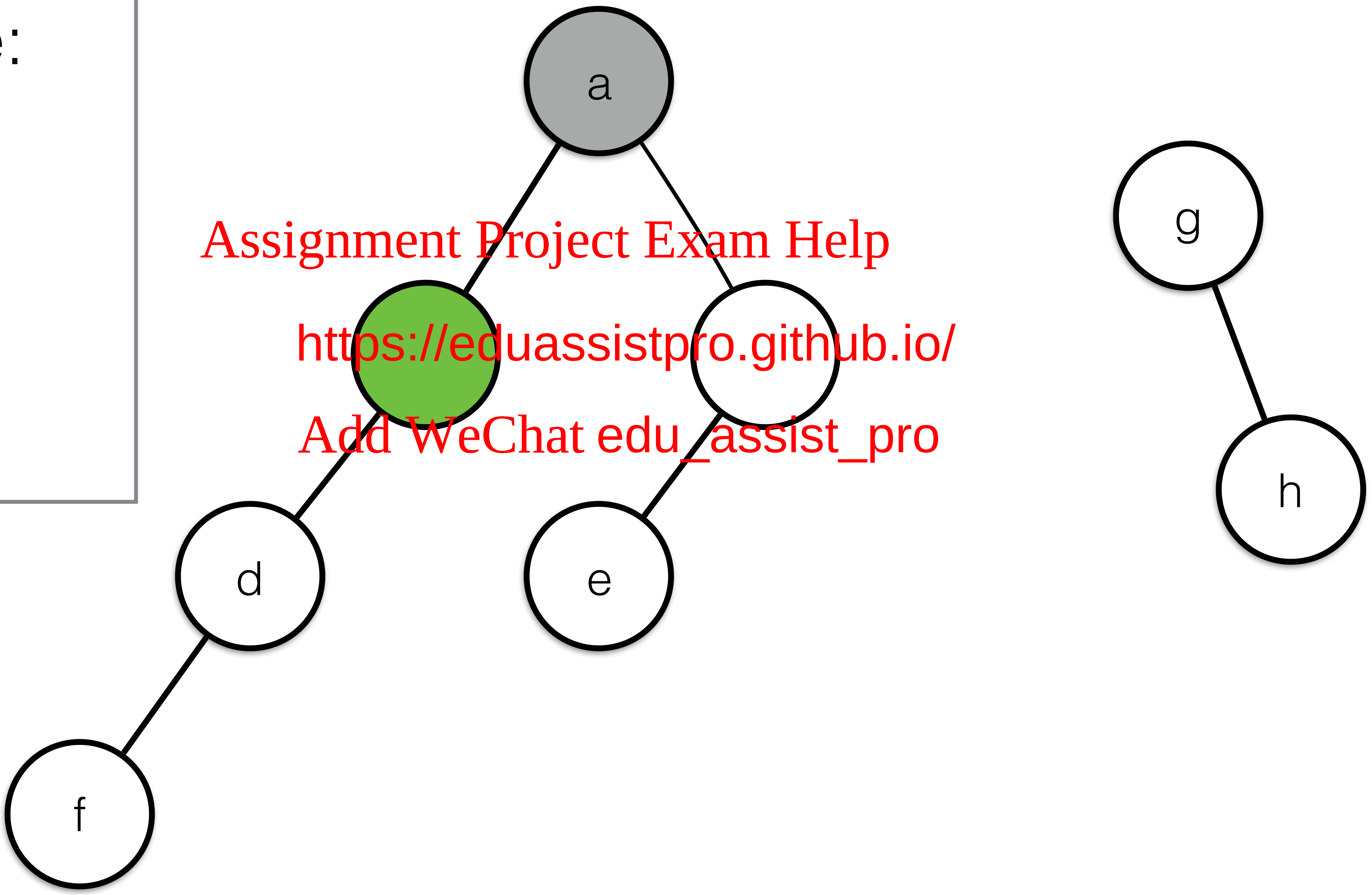


Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal Queue:
c



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

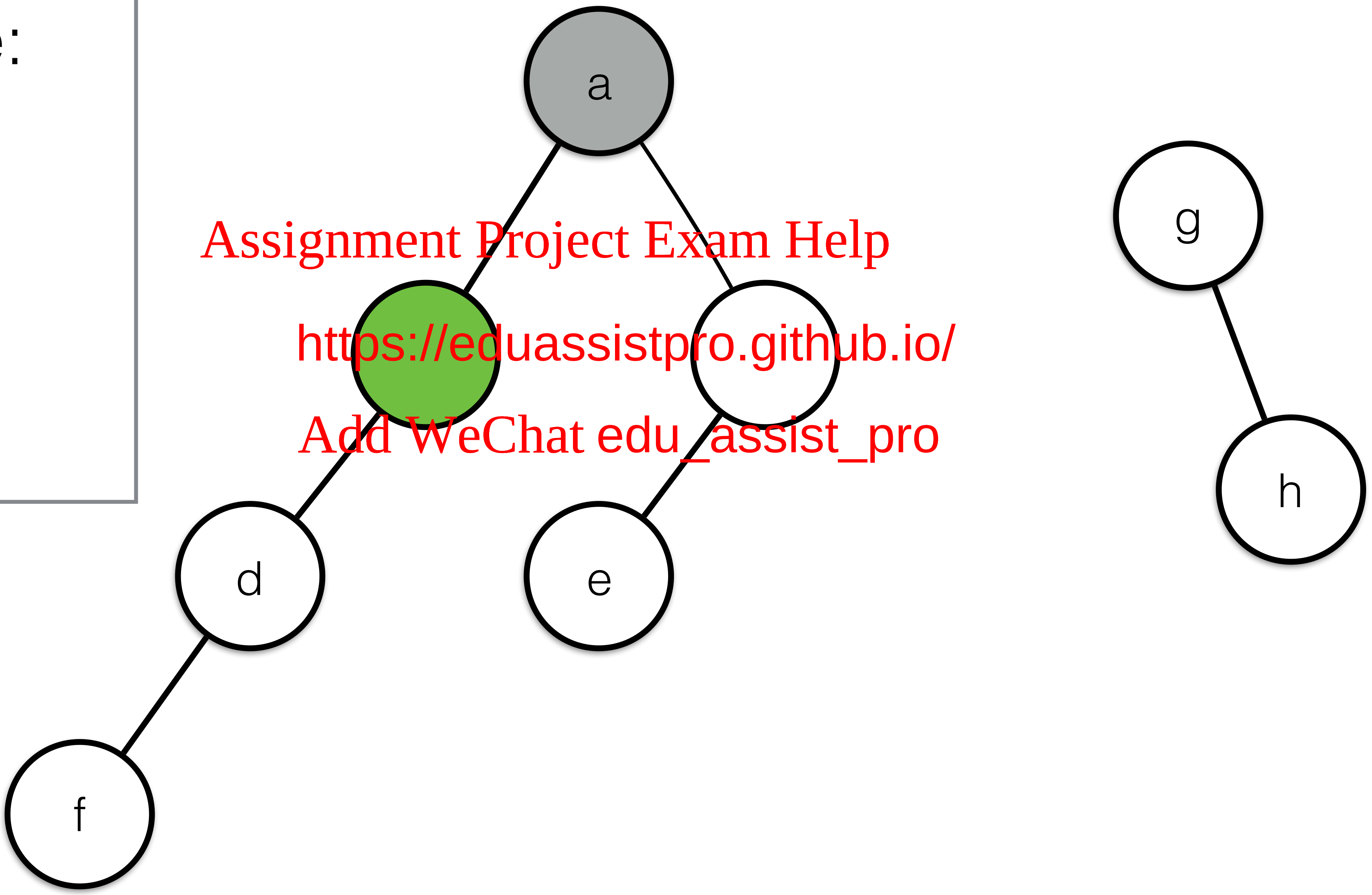
Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal Queue:

c

d



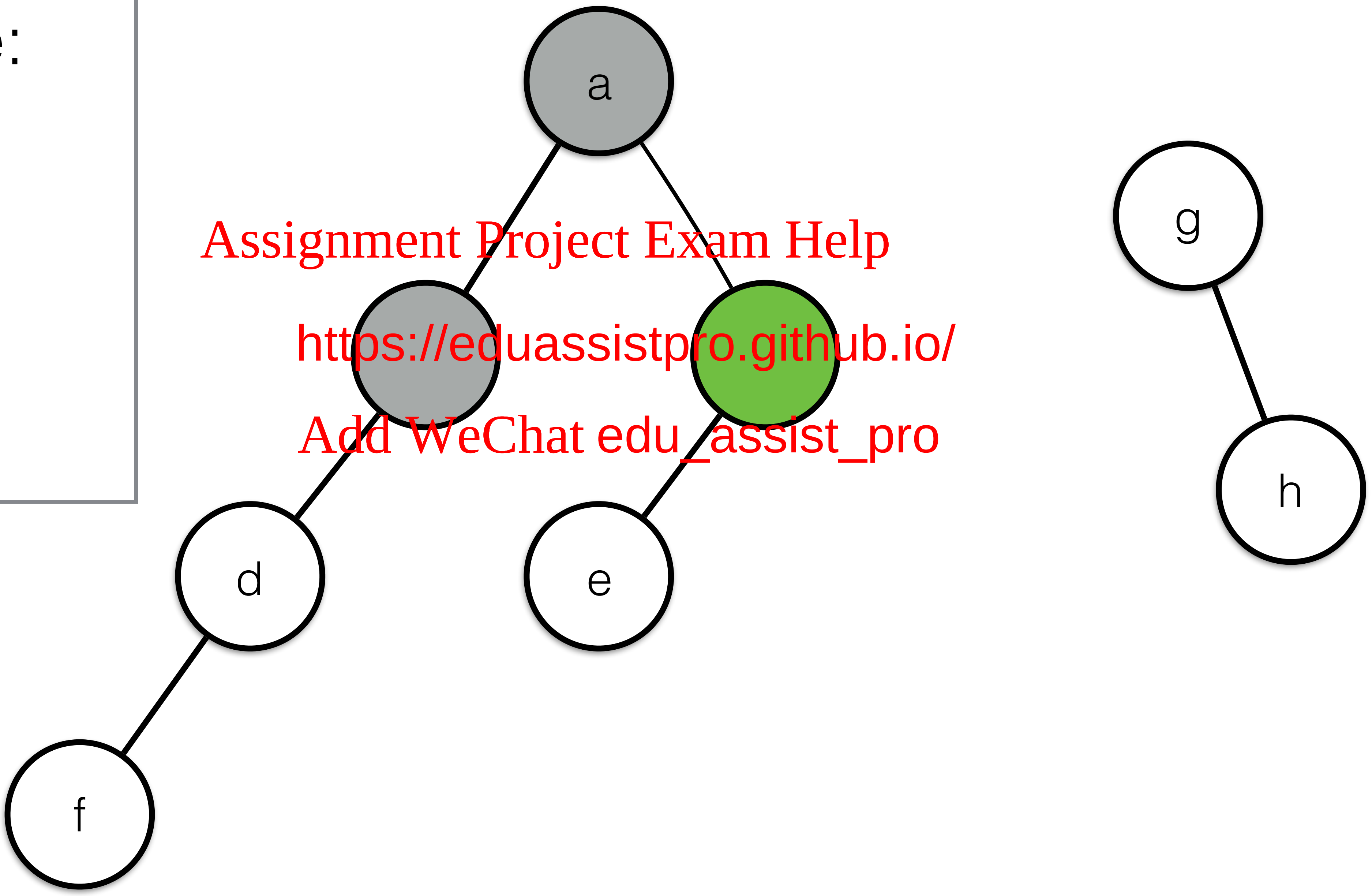
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal Queue:

d



Assignment Project Exam Help

https://eduassistpro.github.io/

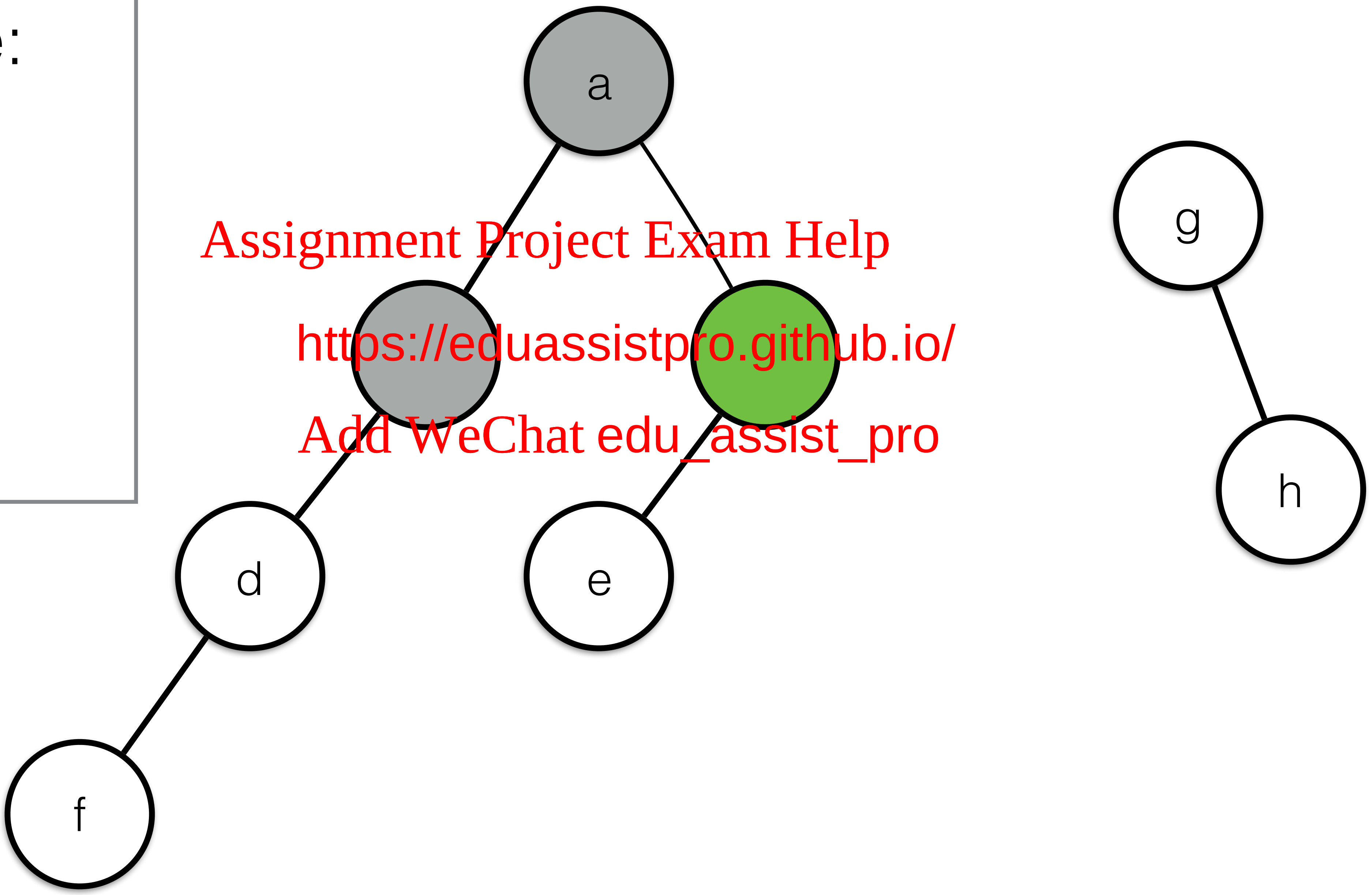
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal Queue:

d
e



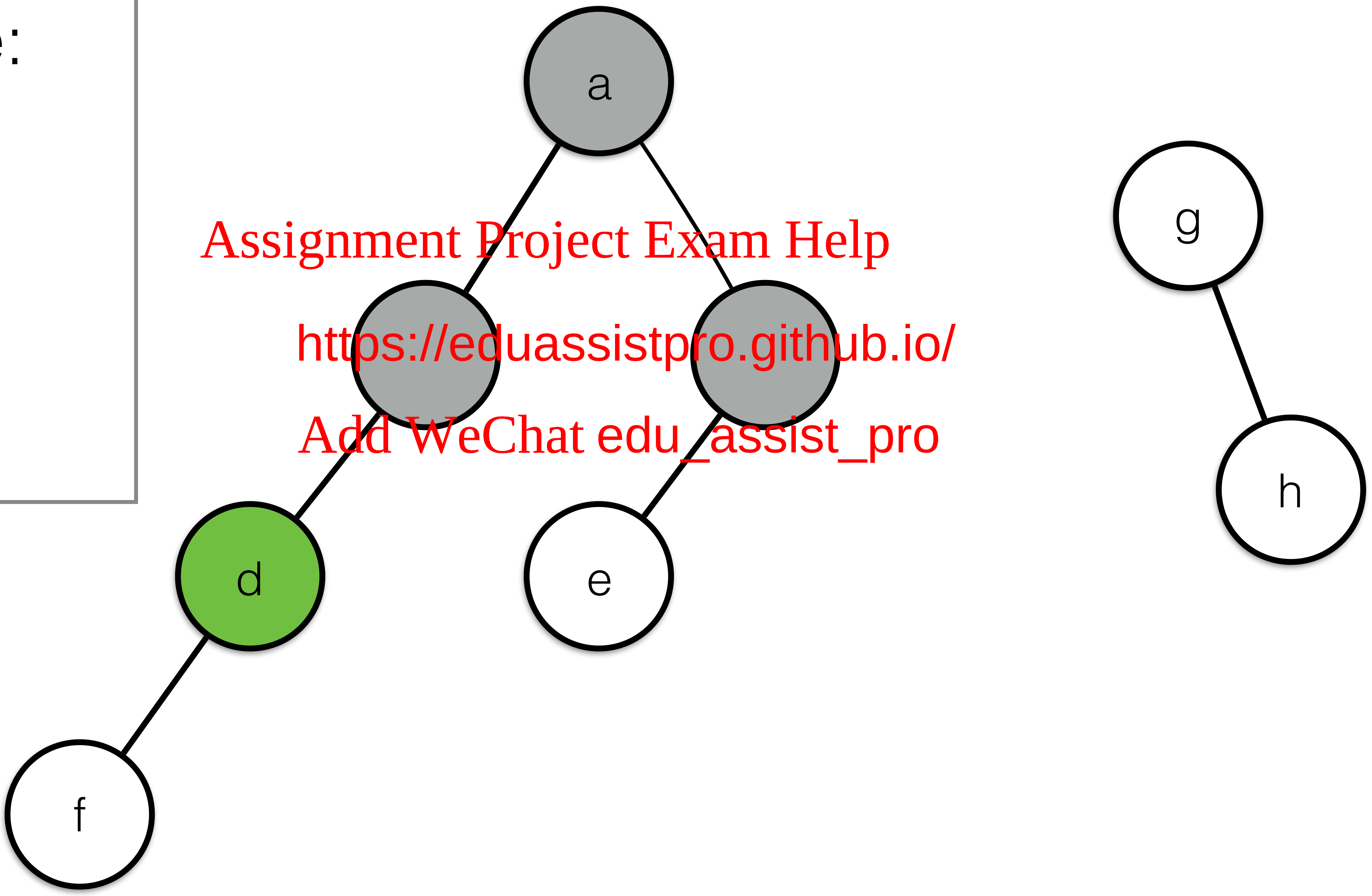
Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal Queue:

e



Assignment Project Exam Help

https://eduassistpro.github.io/

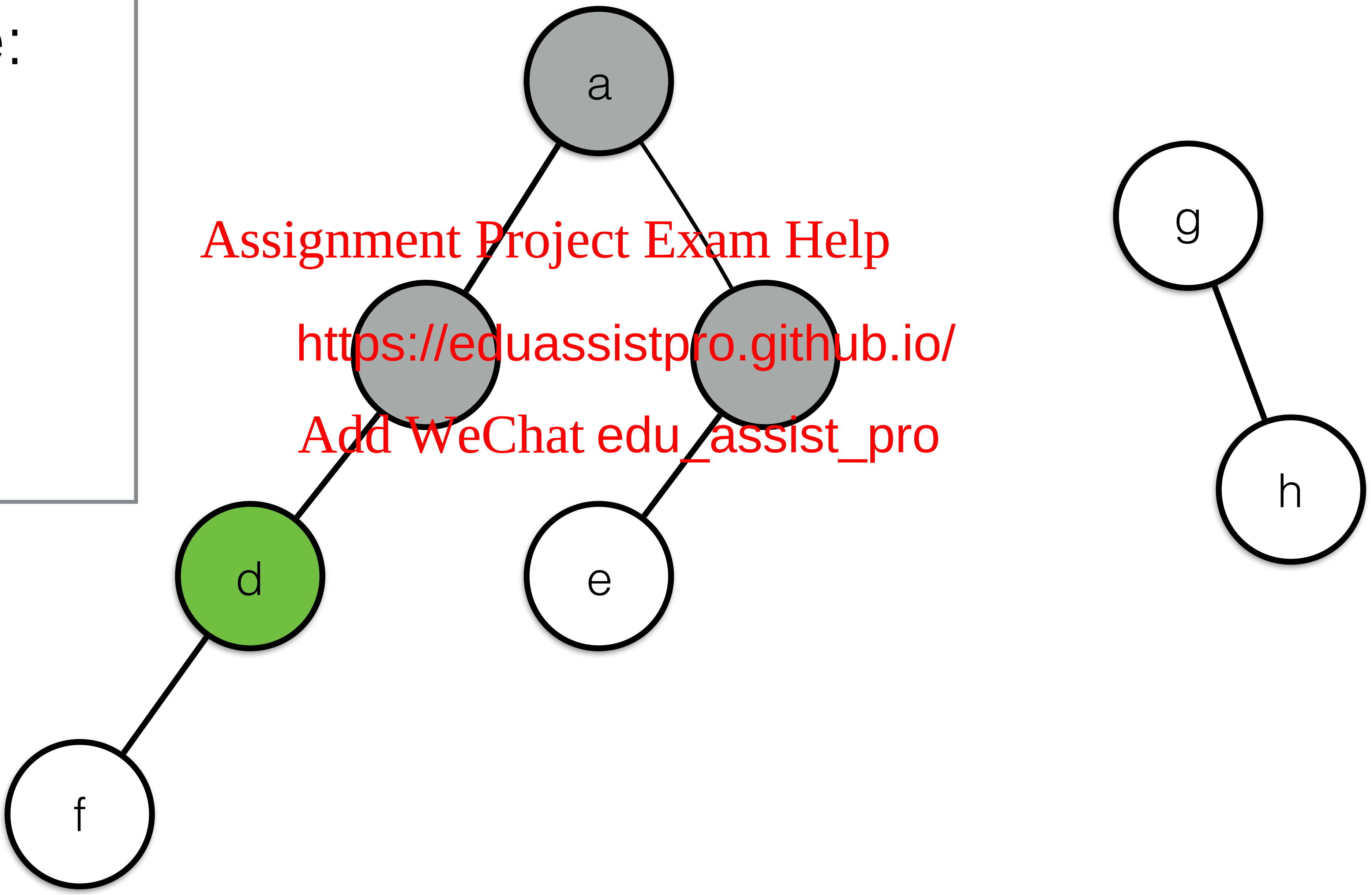
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal Queue:

e
f

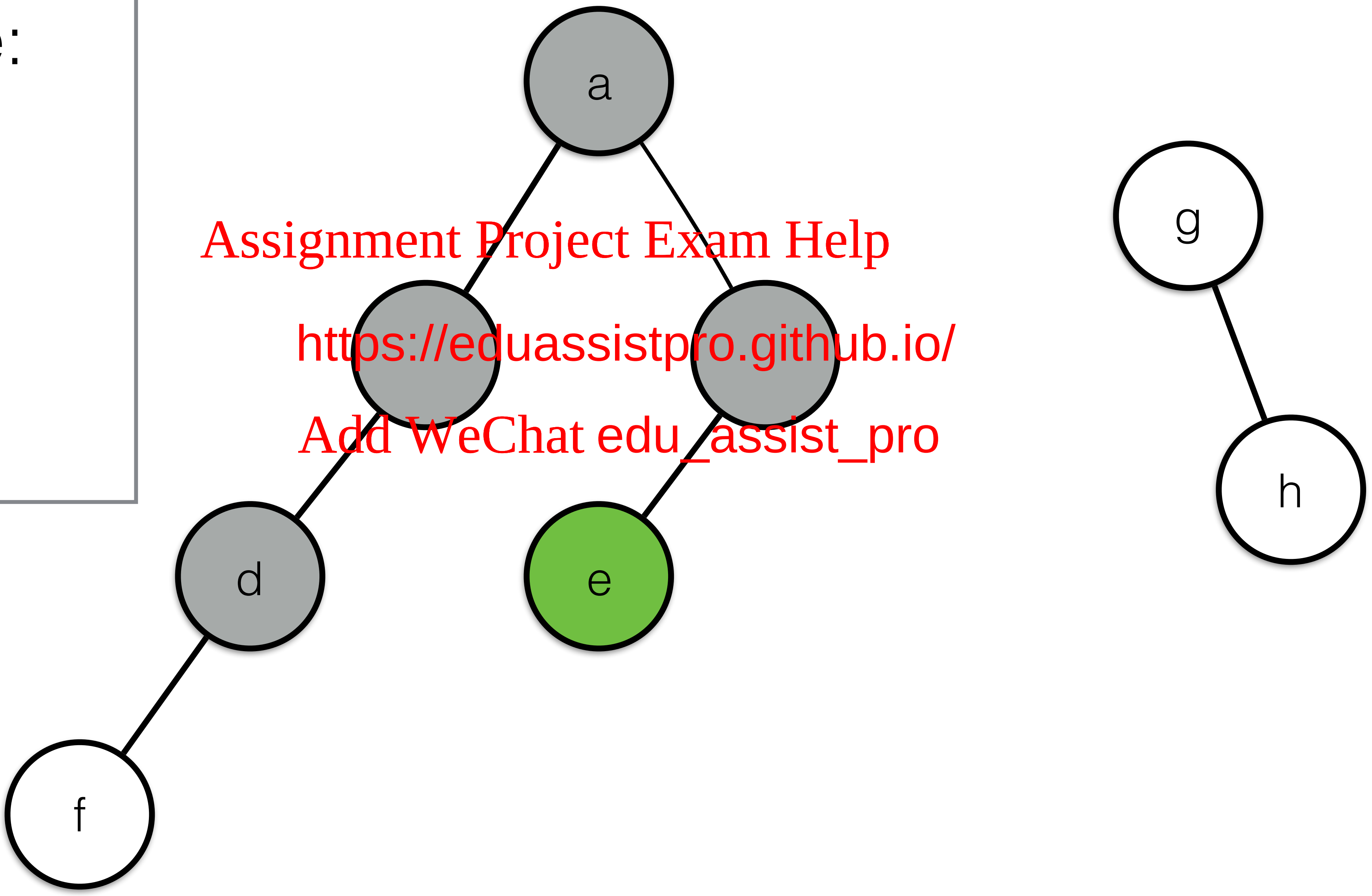


Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal
Queue:
f

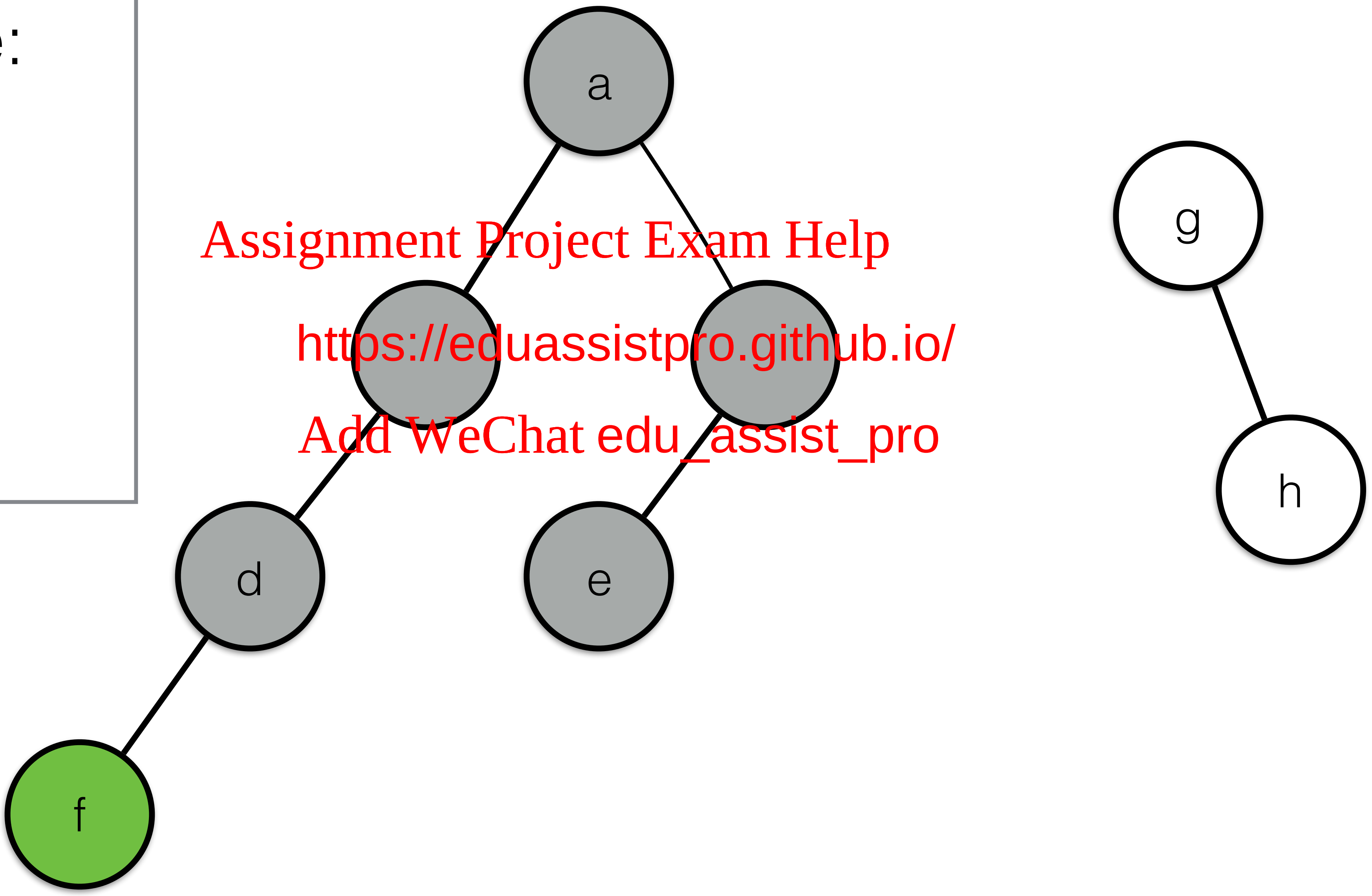


Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal
Queue:

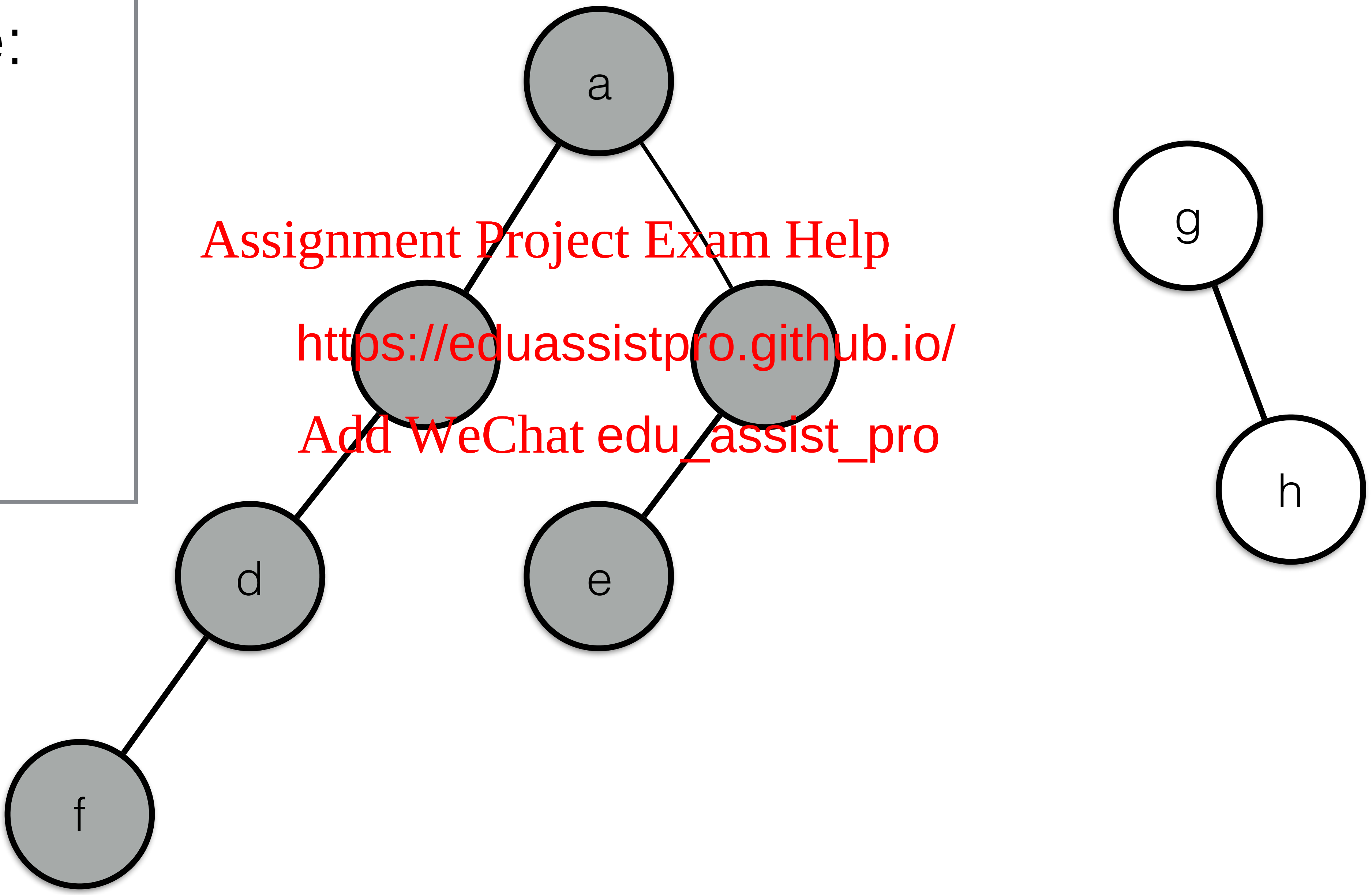


Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal Queue:

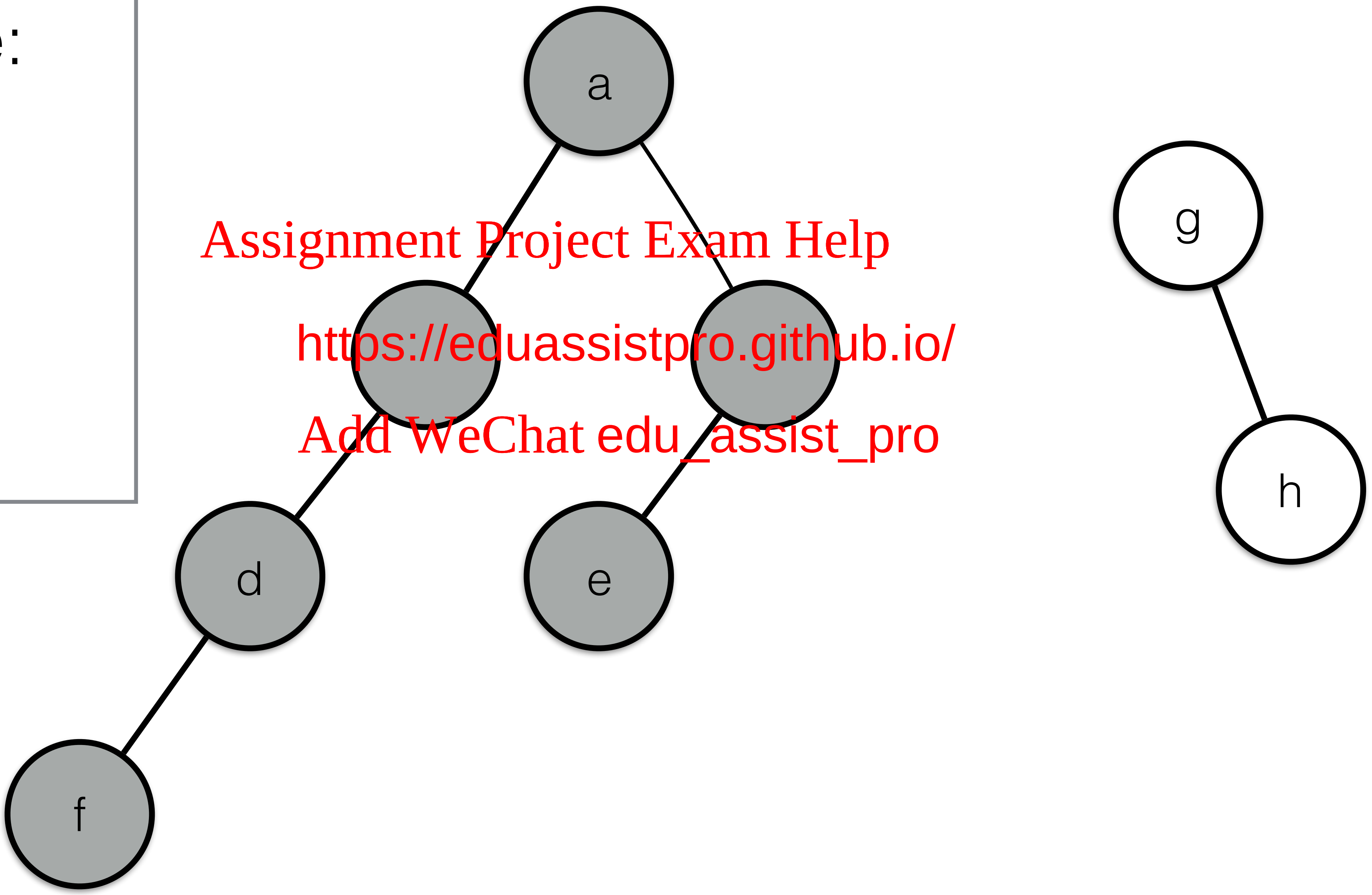


Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal Queue:
g

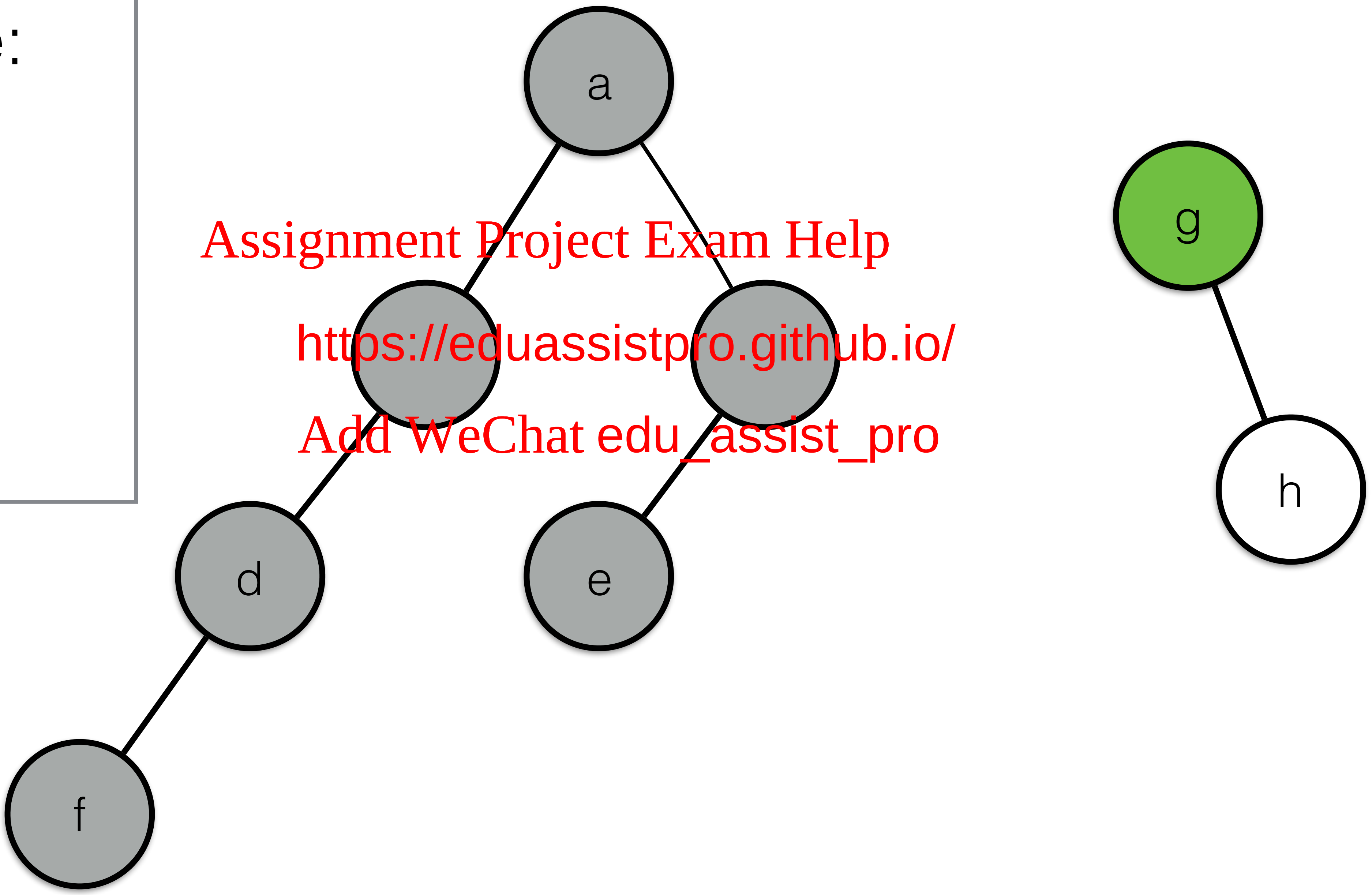


Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal Queue:

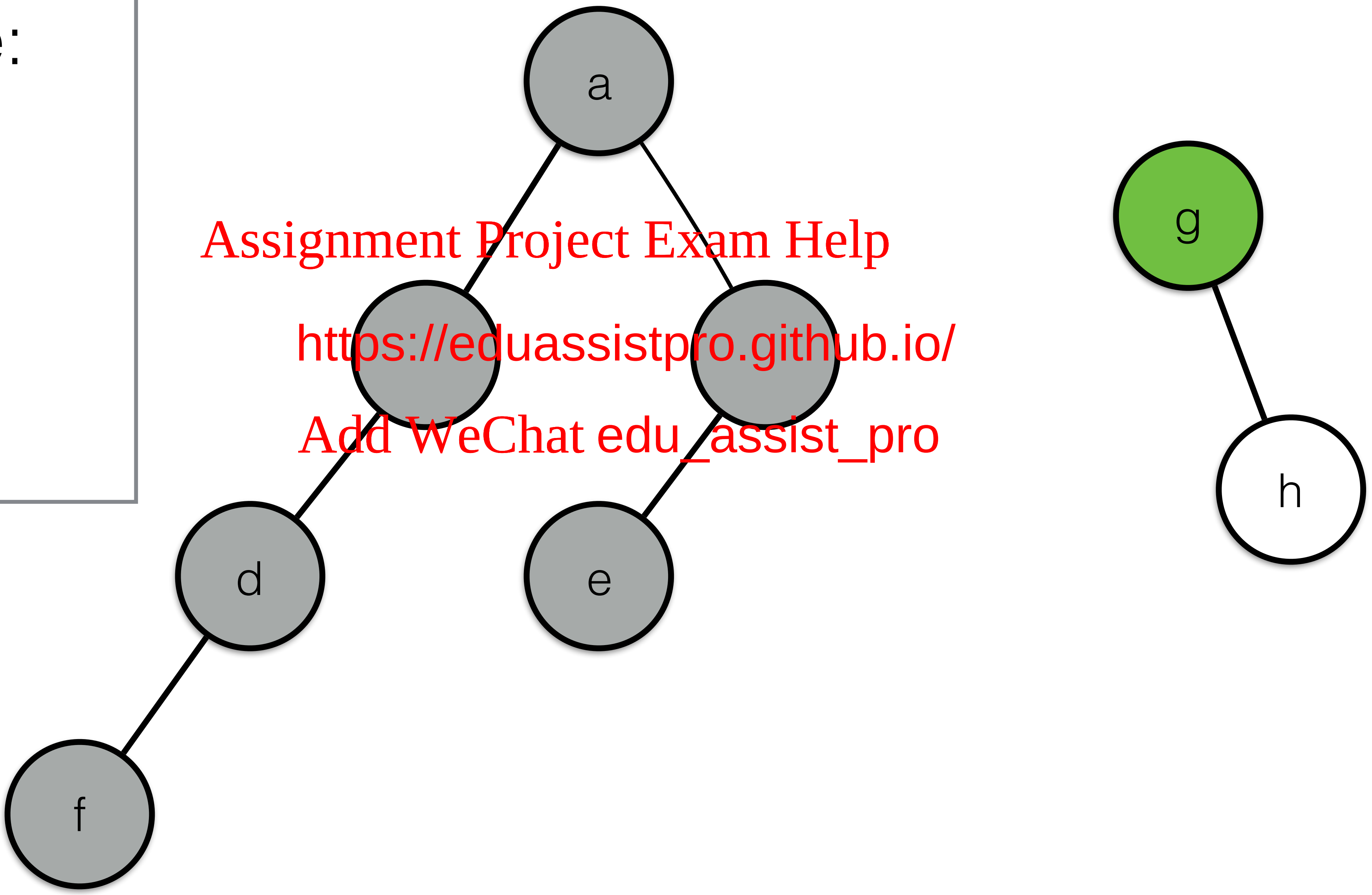


Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal
Queue:
h

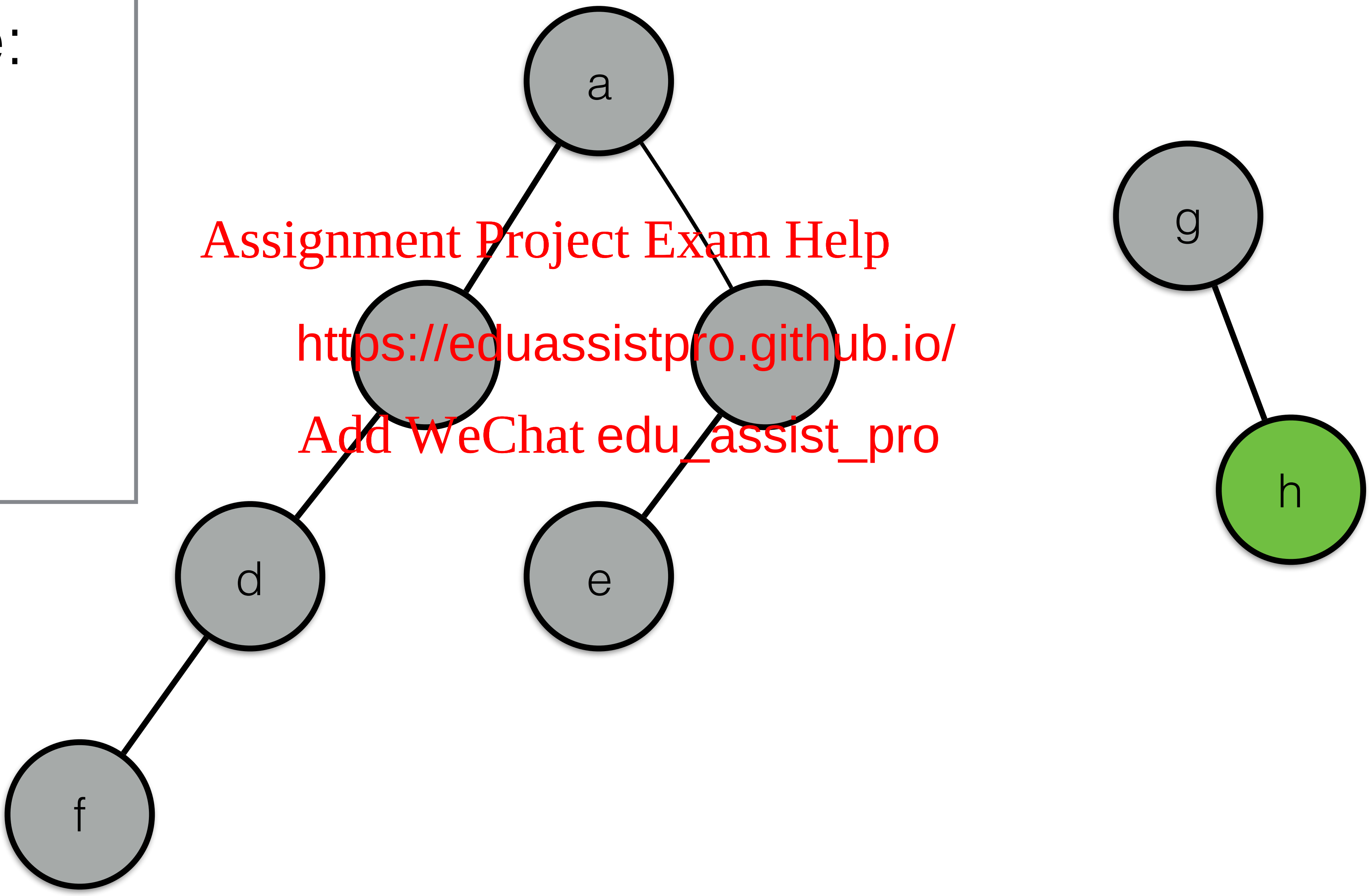


Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search: Queue Discipline

Traversal Queue:



Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search Algorithm

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

BFS Algorithm Notes

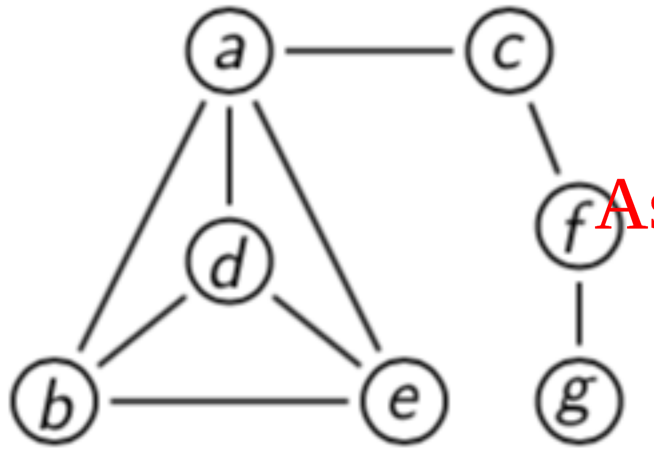
- BFS has the same complexity as DFS.
- Again, the same algorithm works for directed graphs as well.
- Certain problems solved by adapting BFS.
- For example, given a graph and two nodes, a and b in the graph, how would you find the length of the shortest path from a to b?

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Breadth-First Search Forest

BFS **Tree** for this connected graph:



Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

In general, we may get a BFS **Forest**

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Topological Sorting

THE UNIVERSITY OF MELBOURNE

- We mentioned scheduling problems and their representation by directed graphs.
- Assume a directed edge from a to b means that task a must be completed before b can be started.
- The graph must be a d oblem cannot be solved.
- Assume the tasks are carried out b person, unable to multi-task.
- Then we should try to **linearize** the graph, that is, order the nodes as a sequence $v_1, v_2, \ldots, v_n$ such that for each edge $(v_i, v_j) \in E$, we have v_i comes before v_j in the sequence (that is, v_i is scheduled to happen before v_j).

Assignment Project Exam Help
https://eduassistpro.github.io/
Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Topological Sorting Example

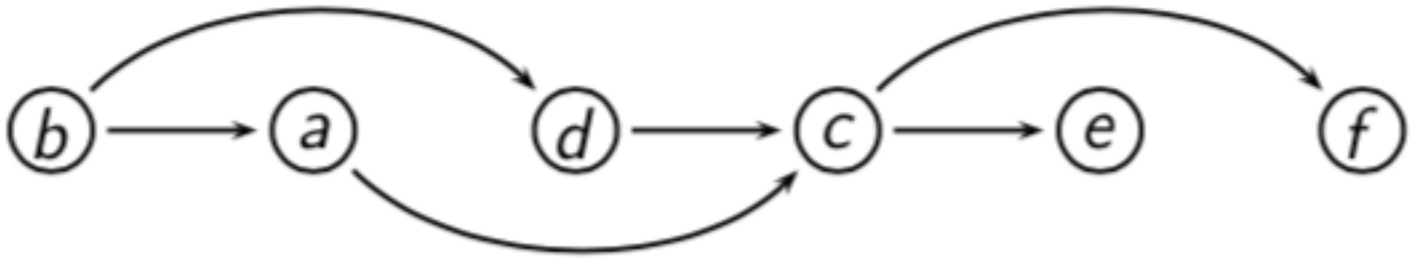
There are 4 ways to linearise the following graph

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Here is one:



$$\begin{tikzcd}
b \arrow[r] \arrow[rr, bend left] & a \arrow[rr, bend right] & d \arrow[r] & c \arrow[r] \arrow[rr, bend left] & e & f
\end{tikzcd}$$

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Topological Sorting Algorithm 1

- We can solve the top-sort problem with depth-first search:

  1. Perform DFS and note the order in which nodes are popped off the stack.

  2. List the nodes in the reverse of that order.

Assignment Project Exam Help

- This works because [illegible]

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

- If (u,v) is an edge then it is possible (given some way of deciding ties) to arrive at a DFS stack with u sitting below v.

- Taking the "reverse popping order" ensures that u is listed before v.

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

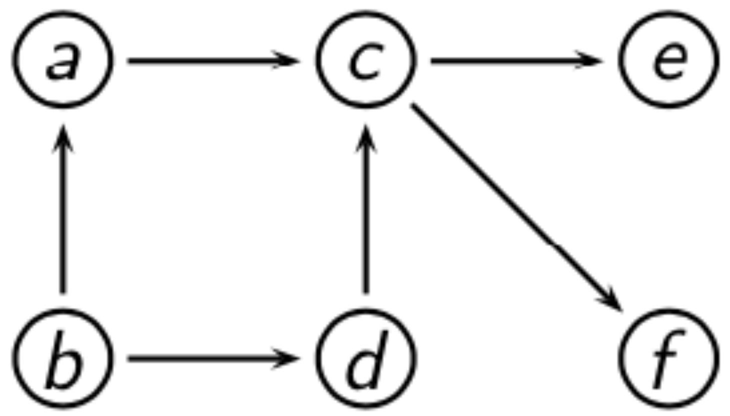
Topological Sorting Example Again

Using the DFS method and resolving ties by using alphabetical order, the graph gives rise to the traversal stack shown on the right (the popping order shown in red):

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



Taking the nodes in reverse popping order yields b, d, a, c, f , e.

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Topological Sorting Algorithm 2

- An alternative method would be to repeatedly select a random **source** in the graph (that is, a node with no incoming edges), list it, and remove it from the graph (including removing its outgoing edges).

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

- This is a very natural approach, but it has the drawback that we repeatedly need to scan the graph for a source.

- However, it exemplifies the general principle of **decrease-and-conque**r.

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

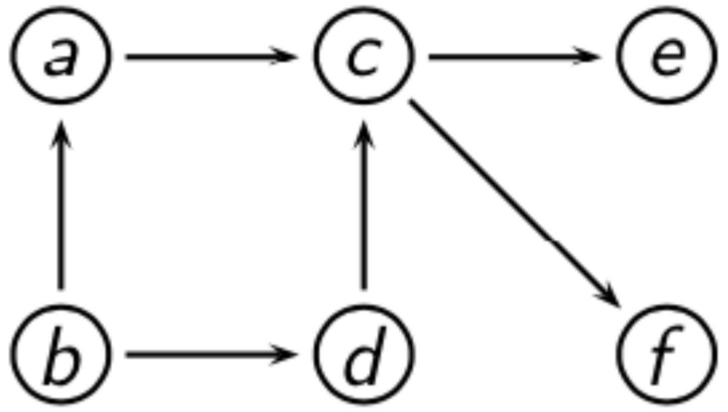
Topological Sorting Example Again

Using the source removal method (and resolving ties alphabetically):

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro



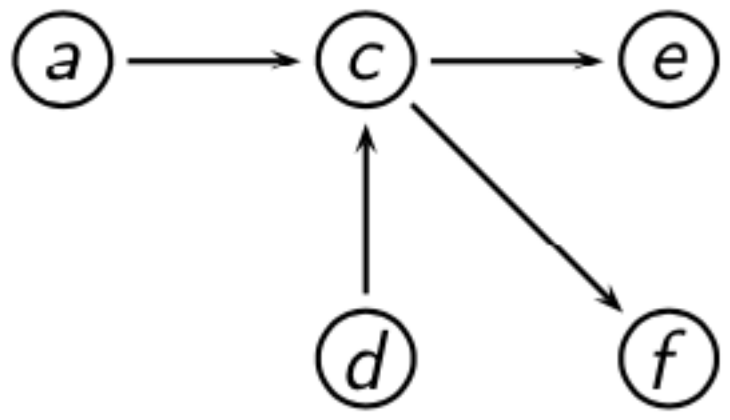
Topological sorted order:

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Topological Sorting Example Again

Using the source removal method (and resolving ties alphabetically):

Assignment Project Exam Help



https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Topological sorted order:

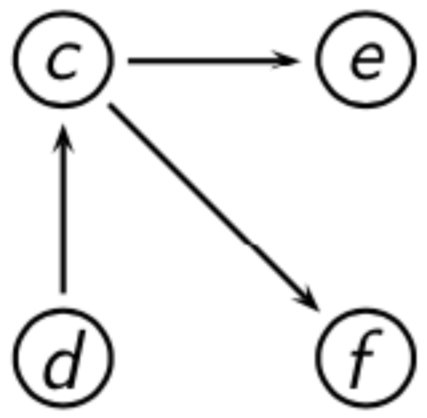
b

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Topological Sorting Example Again

Using the source removal method (and resolving ties alphabetically):

Assignment Project Exam Help



https://eduassistpro.github.io/

Add WeChat edu_assist_pro

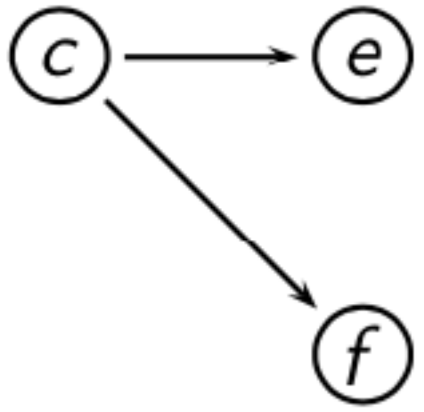
Topological sorted order:
b, a

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Topological Sorting Example Again

Using the source removal method (and resolving ties alphabetically):

Assignment Project Exam Help



https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Topological sorted order:
b, a, d

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Topological Sorting Example Again

Using the source removal method (and resolving ties alphabetically):

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Topological sorted order:

b, a, d, c

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Topological Sorting Example Again

Using the source removal method (and resolving ties alphabetically):

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Topological sorted order:
b, a, d, c, e

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Topological Sorting Example Again

Using the source removal method (and resolving ties alphabetically):

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Topological sorted order:
b, a, d, c, e, f

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License

Next time

THE UNIVERSITY OF MELBOURNE

Assignment Project Exam Help

- So next we turn o e very useful "decrease and c e (Levitin Chapter 4).

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Copyright University of Melbourne 2016, provided under Creative Commons Attribution License