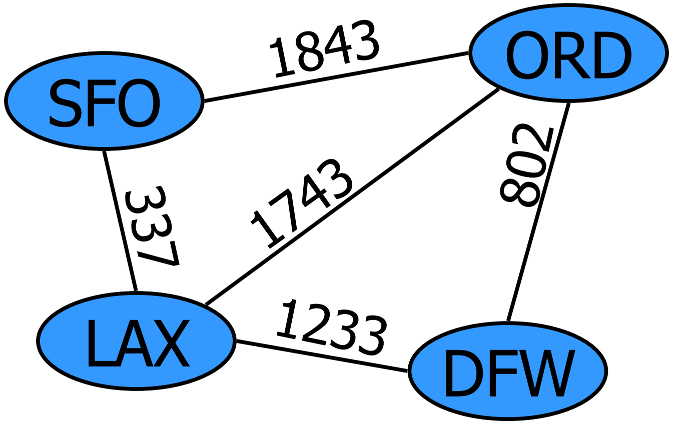
Graph Traversal (BFS)



Instructor: Samreen Ishfaq

Breadth-First Search

❖Start from a given vertex *v*.
❖Visit all neighbors of *v*.
❖Then visit all neighbors of first neighbor *w* of *v*.
❖Then visit all neighbors of second neighbor *x* of *v*, etc.

❖*BFS visits nodes level by level.*
While visiting each node on a given level,
store it , so that,

✓we can return to it after completing this level
✓so that nodes adjacent to it can be visited.

❖Because the first node visited on a given level should be the first one to which we return, a *queue* is an appropriate data structure for storing the nodes.

Breadth-First Search

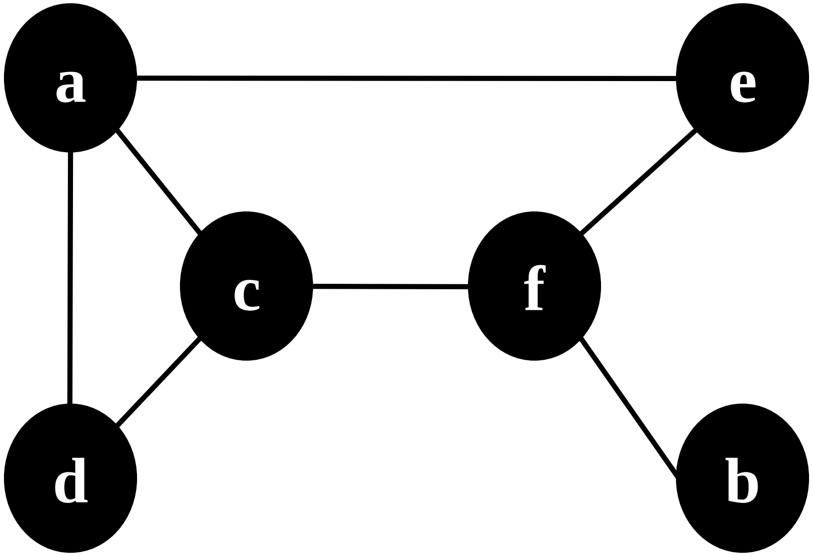
```
BFS(G)
mark each vertex with -1
for each vertex v∈ V do
        bfs(v)
----------------------------

bfs(v)
mark v with 1
initialize queue with v
while queue is not empty do
        a := remove  front element of the queue
        for each vertex w adjacent to a do
                if w is marked with -1
                        mark w with 1
                        add w to the end of the queue
```

Breadth-First Search

a	b	c	d	e	f
-1	**-1**	**-1**	**-1**	**-1**	**-1**

BFSnumber

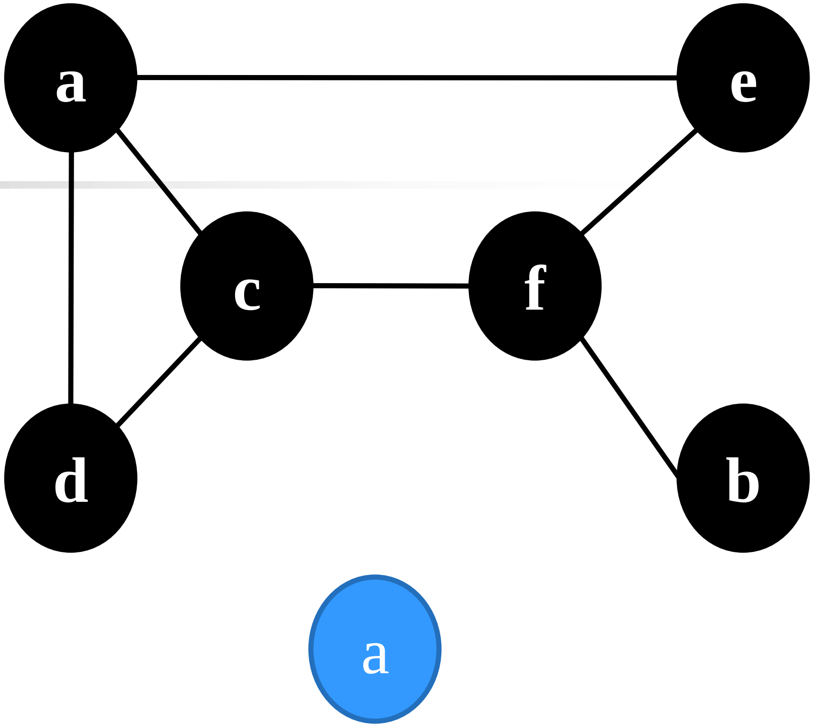


Queue:

Breadth-First Search

a	b	c	d	e	f
1	-1	-1	-1	-1	-1

BFSnumber



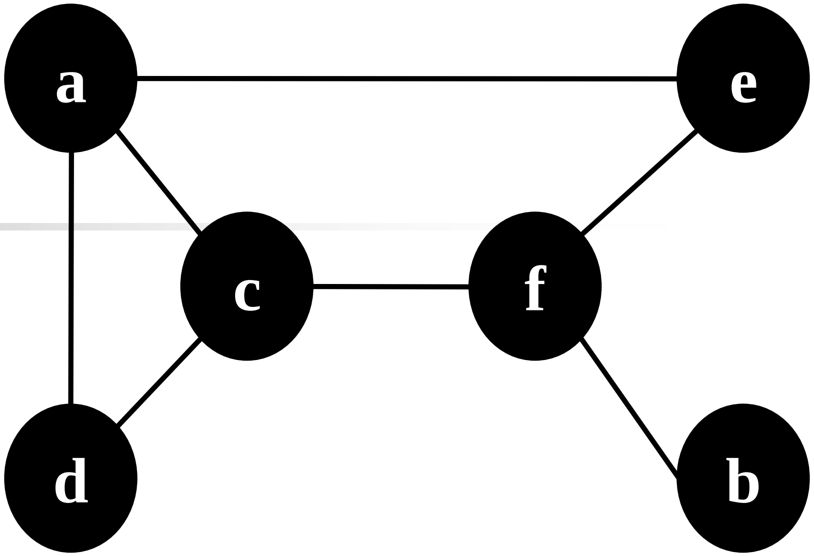
Queue:

a

front

rear

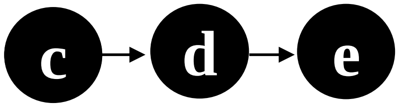
Breadth-First Search



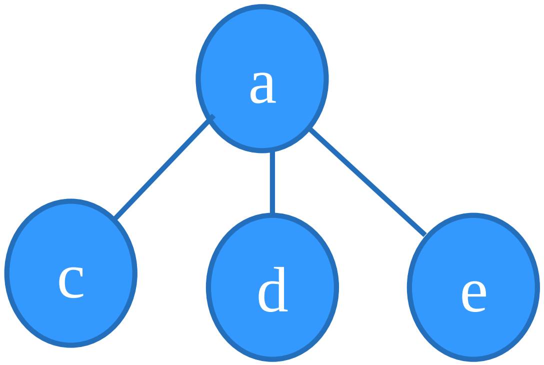
a	b	c	d	e	f
1	-1	1	1	1	-1

BFSnumber

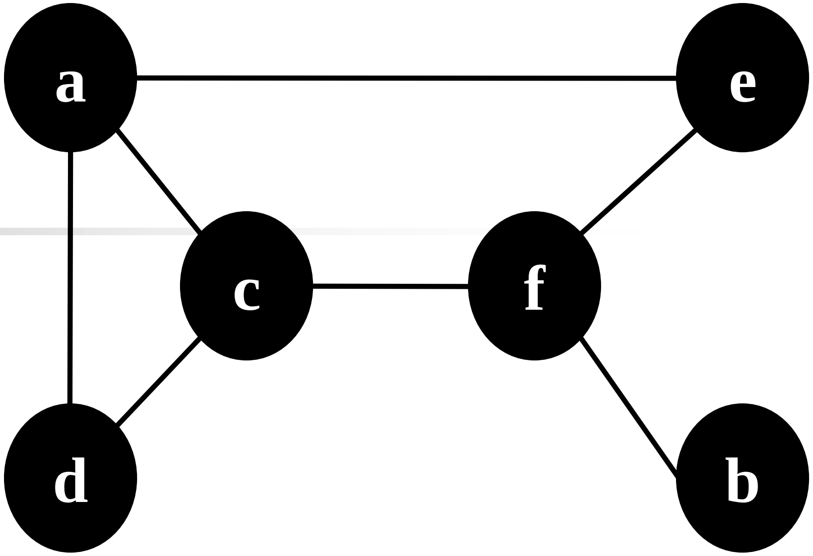
Queue:



front rear



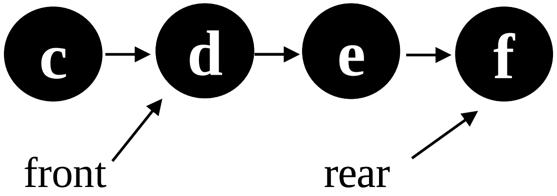
Breadth-First Search

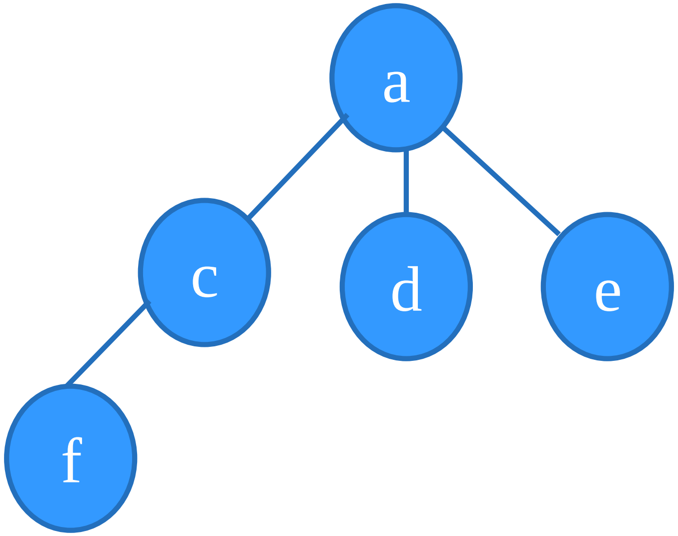


a	b	c	d	e	f
1	-1	1	1	1	1

BFSnumber

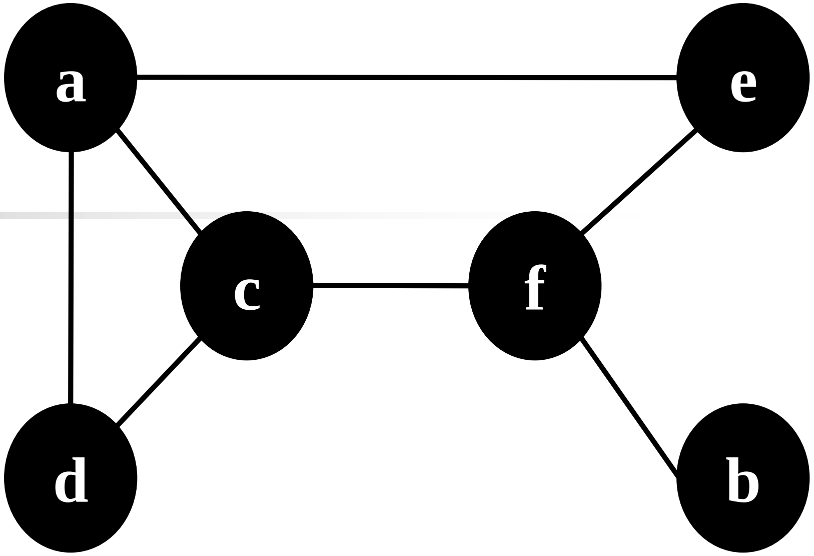
Queue:





Instructor: Samreen Ishfaq

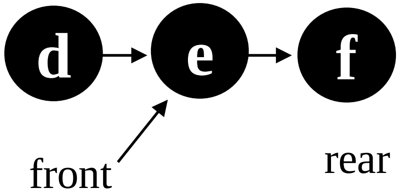
Breadth-First Search

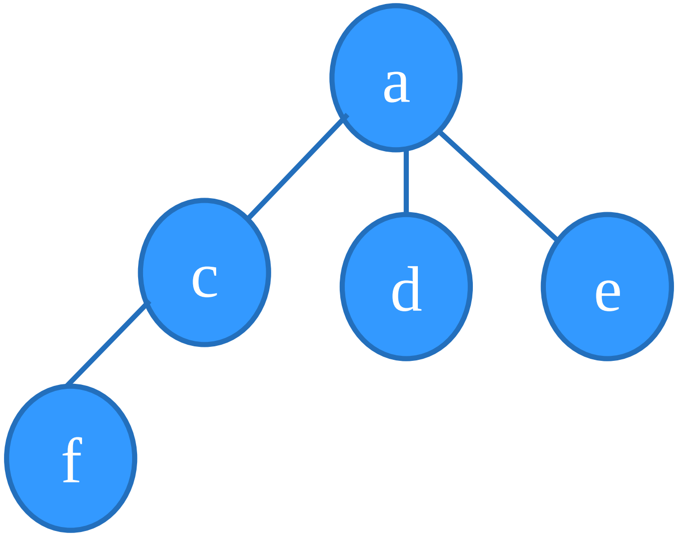


a	b	c	d	e	f
1	-1	1	1	1	1

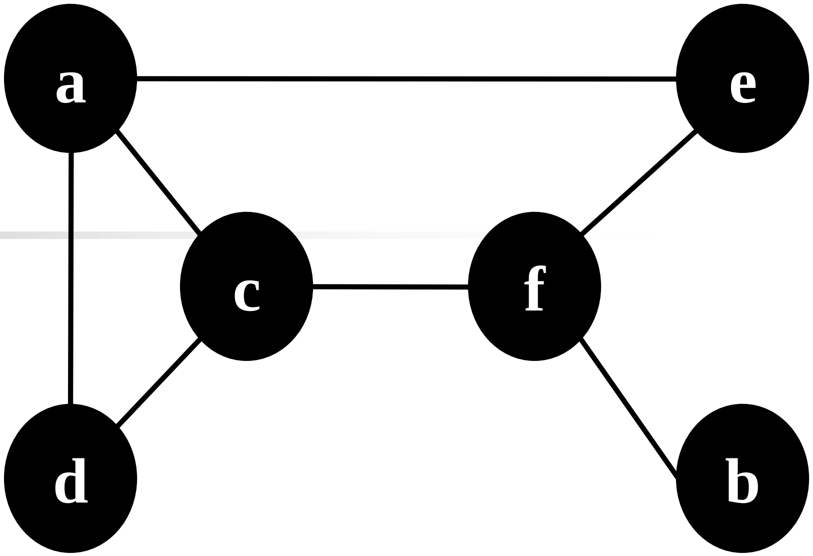
BFSnumber

Queue:





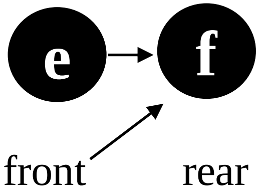
Breadth-First Search

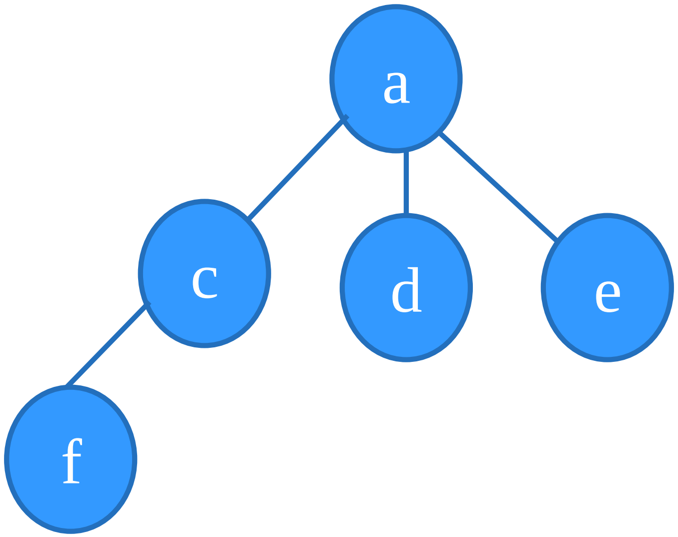


a	b	c	d	e	f
1	-1	1	1	1	1

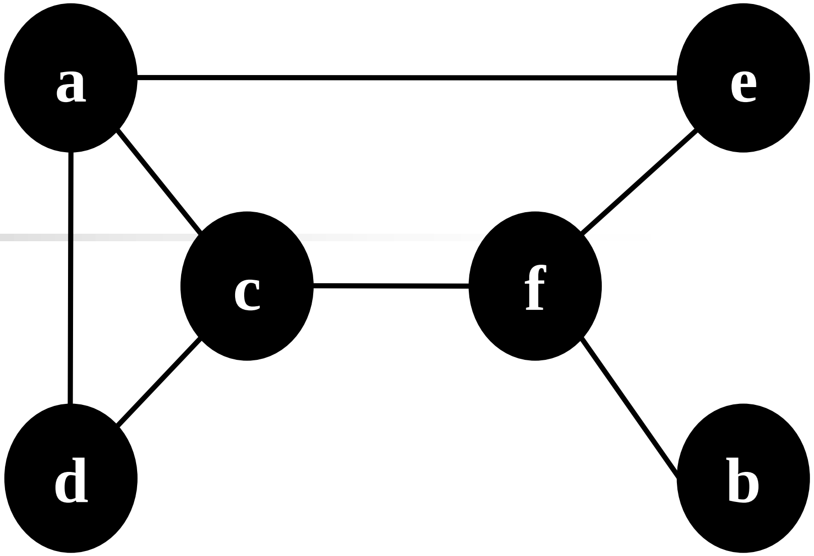
BFSnumber

Queue:





Breadth-First Search



a	b	c	d	e	f
1	1	1	1	1	1

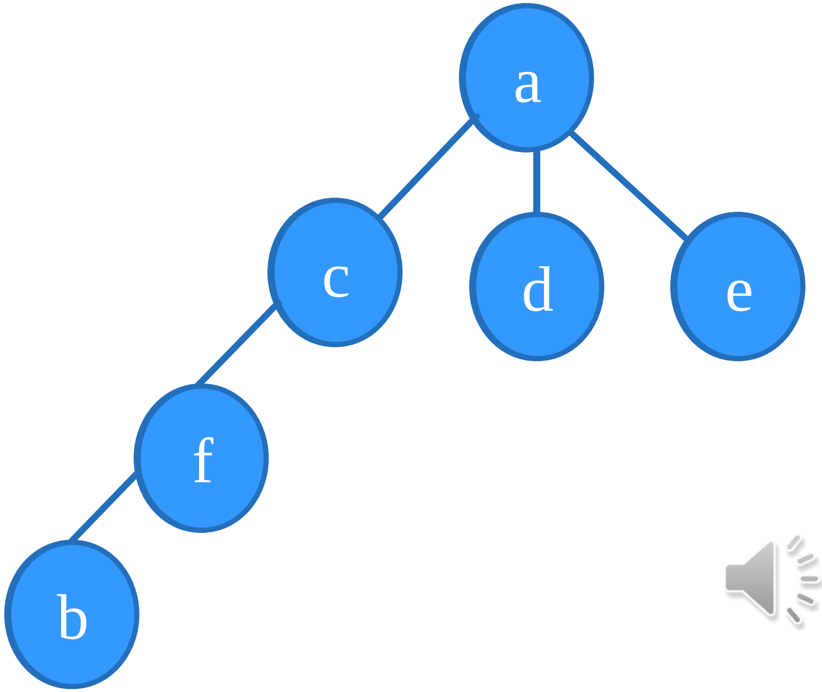
BFSnumber

Queue:

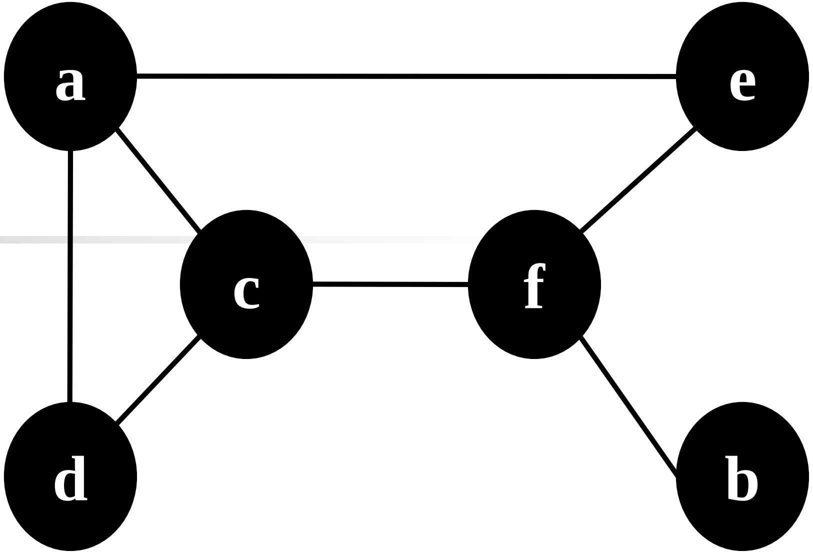
b

front

rear



Breadth-First Search



a	b	c	d	e	f
1	1	1	1	1	1

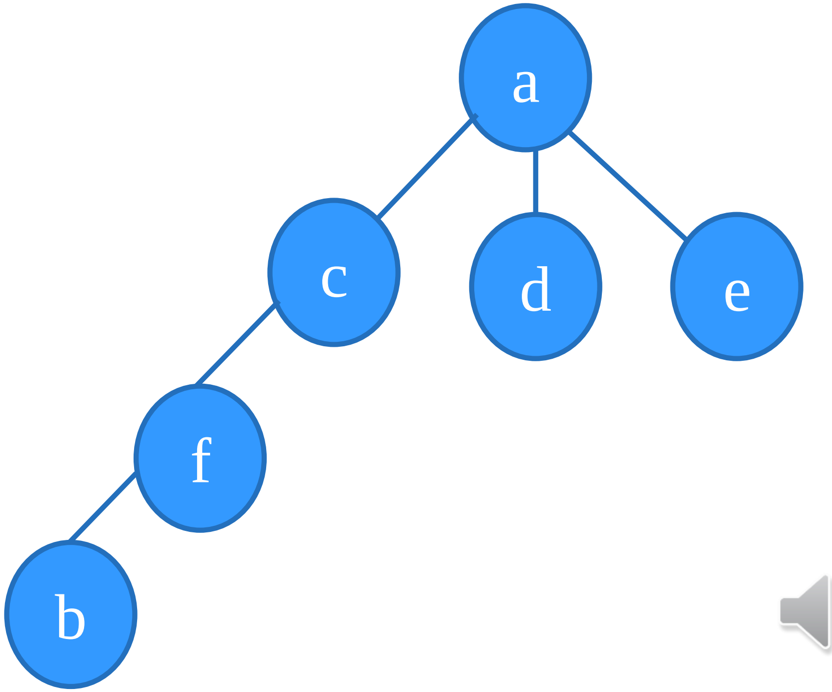
BFSnumber

Queue:

b

front

rear



DFS vs BFS

- Data structures: stack vs. queue
- Implementation: recursion vs. explicit queue manipulation
- BFS is like level order traversal and DFS of a tree would be equivalent to a preorder traversal.