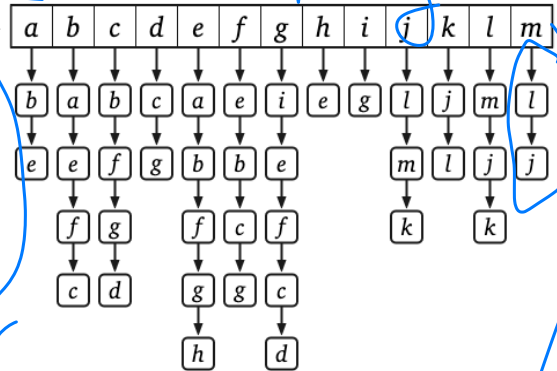
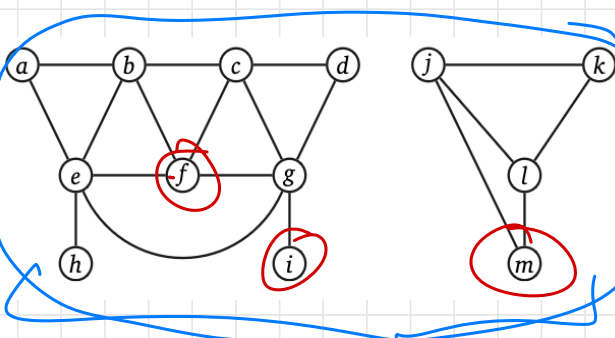


Assume that graphs are represented as standard adjacency lists — array of linked lists



V, vertices
E, edges

Depth-First-Search

RECURSIVEDFS(v):

if v is unmarked

mark v

for each edge vw

RECURSIVEDFS(w)

Breadth-First-Search

ITERATIVE^BDFS(s):

PUSH(s)

while the ~~stack~~ ^{queue} is not empty

$v \leftarrow$ ~~pop~~ ^{dequeue}

if v is unmarked

mark v

for each edge vw

PUSH(w)

WHATEVERFIRSTSEARCH(s):

put s into the bag

while the bag is not empty

take v from the bag

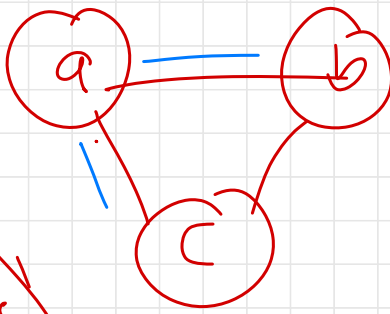
if v is unmarked

mark v

for each edge vw

put w into the bag

bag:
insert
remove
isempty



WFS(a)

bag: when we remove, take alphabetically smallest

bag can contain dups.

after iteration	took out of bag	Bag	verts marked
0	X	a	X
1	a	b c	a

WFSALL(G):

for all vertices v

unmark v

for all vertices v

if v is unmarked

WHATEVERFIRSTSEARCH(v)

COUNTCOMPONENTS(G):

count $\leftarrow 0$

for all vertices v

unmark v

for all vertices v

if v is unmarked

count $\leftarrow$ *count* + 1

WHATEVERFIRSTSEARCH(v)

return count

COUNTANDLABEL(G):

count $\leftarrow 0$

for all vertices v

unmark v

for all vertices v

if v is unmarked

count $\leftarrow$ *count* + 1

LABELONE(v , *count*)

return count

⟨⟨Label one component⟩⟩

LABELONE(v , *count*):

while the bag is not empty

take v from the bag

if v is unmarked

mark v

comp(v) $\leftarrow$ *count*

for each edge vw

put w into the bag

- Constructing Spanning Trees

WHATEVERFIRSTSEARCH(s):

put $(\emptyset, s)$ in bag

while the bag is not empty

take (p, v) from the bag

(*)

if v is unmarked

mark v

$\text{parent}(v) \leftarrow p$

for each edge vw

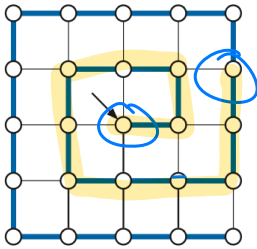
(†)

put (v, w) into the bag

(**)

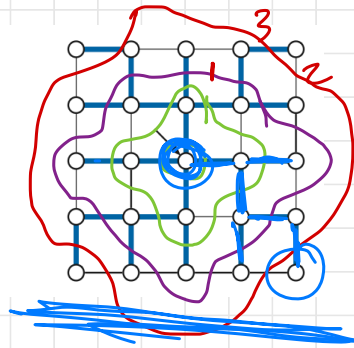
assign a parent to every vertex
reachable from s (except s itself)

the edges $v \rightarrow v.\text{parent}$ form a



stack

→ DFS



queue

→ BFS

⇒ shortest paths

spanning
tree of
reachable
verts

if G is weighted (with positive ints)
what sort of bag to compute:

→ ① weighted shortest paths
(Dijkstra's alg)

② minimum weight spanning tree
(Prim's alg)

Dijkstra's alg
smallest — priority queue

WHATEVERFIRSTSEARCH(s):

put s into the bag

while the bag is not empty

take v from the bag

if v is unmarked

mark v

for each edge vw

put w into the bag