

Lecture 19 Scribbles
Chat Moderators: Vasilis & Samir

Topics: - Breadth First Search
- Dijkstra's Algorithm

DFS good for exploring graph structure

BFS good for exploring distances

Remember Queues (FIFO)

- enqueue to add element to end

- dequeue to remove element from beginning

BFS(s) $O(|V| + |E|)$
 $O(n + m)$

Mark all v as unvisited

for each $v \in V$ $dist(v) = \infty$

Initialize $T = \{\}$

Initialize Q as empty

Q . Enqueue (s)

Mark s as visited, $dist(s) = 0$

While Q is not empty

$u = Q$. dequeue

For each v adjacent to u

if v is not visited

add (u, v) to T

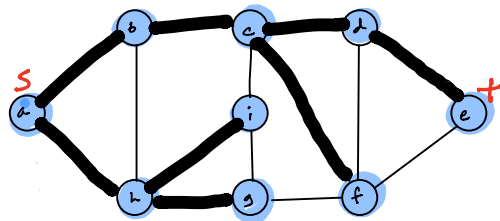
Mark v as visited

Enqueue v

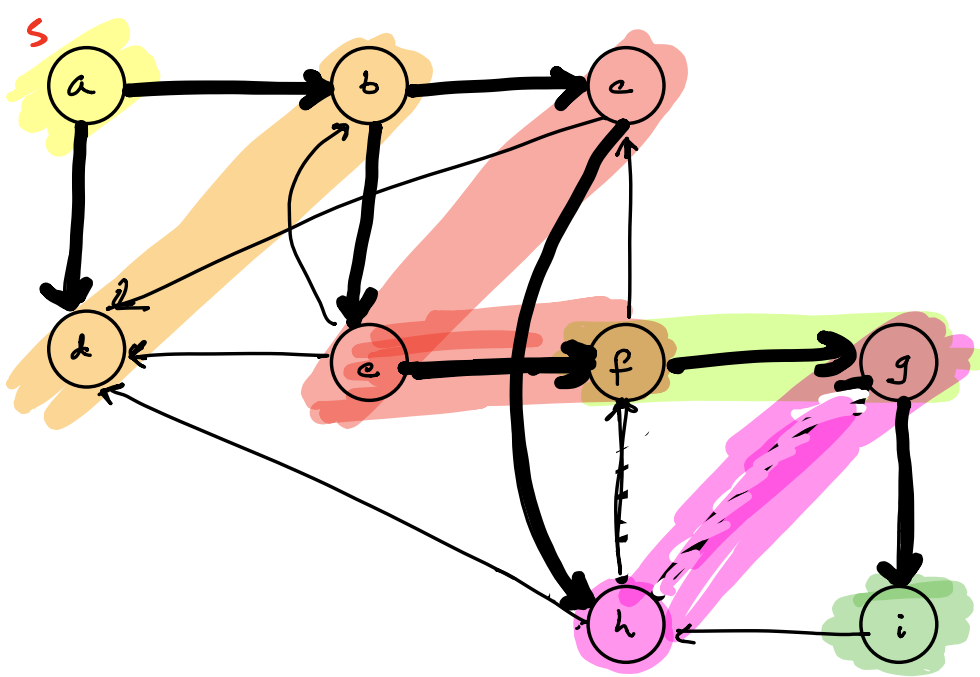
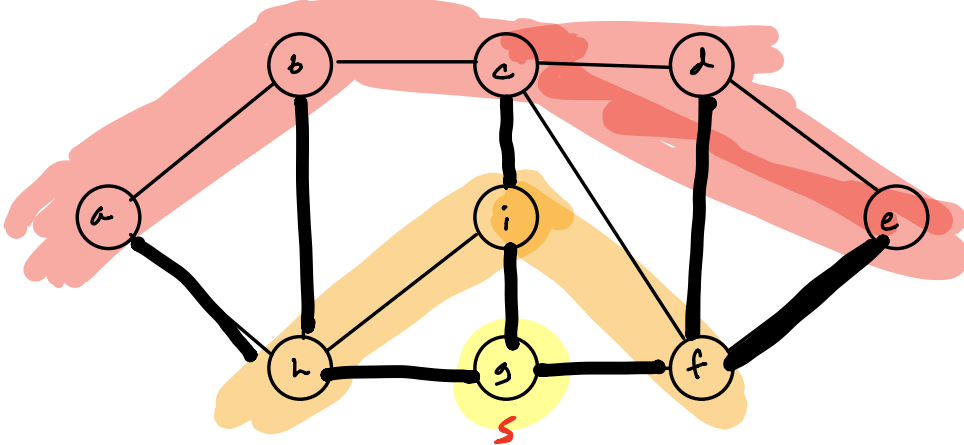
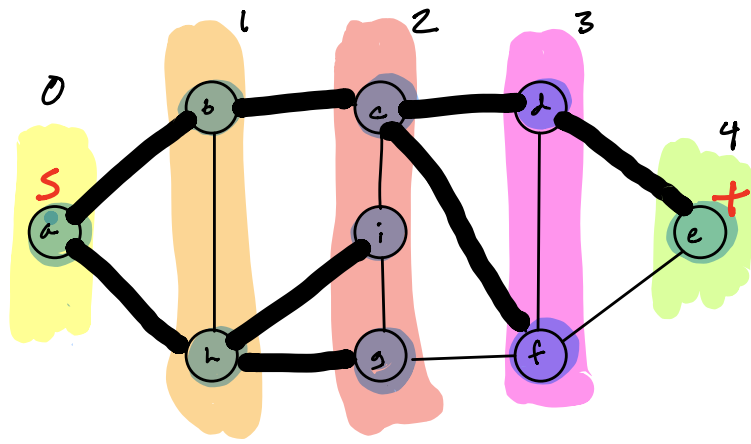
$dist(v) = dist(u) + 1$

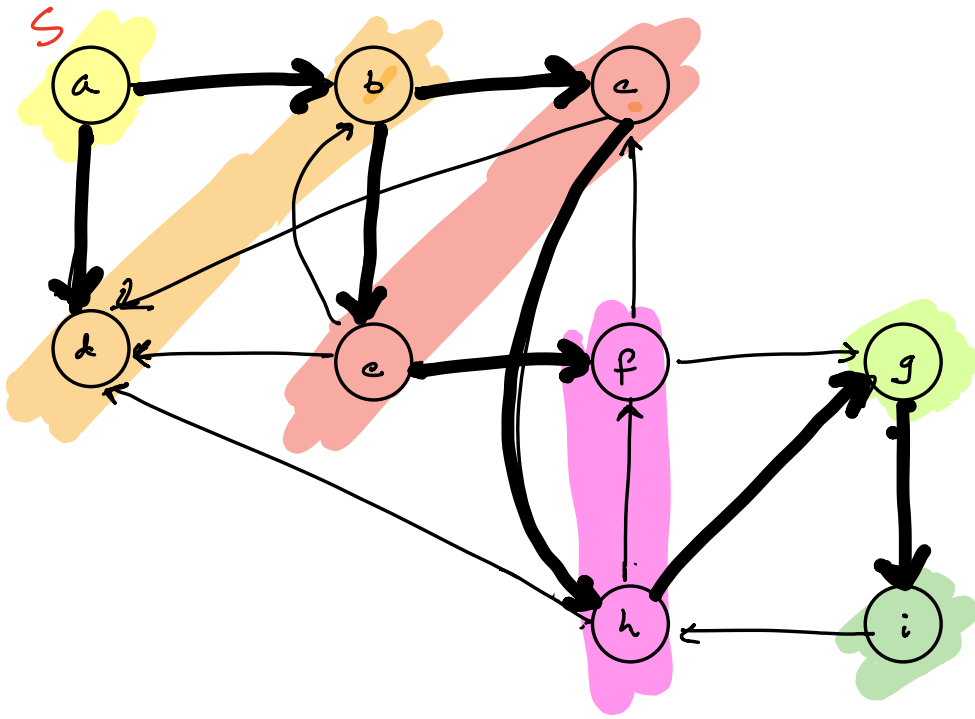
each edge $dist(e) = 1$

↪ search tree



Q
~~a~~
~~b~~
~~h~~
~~e~~
~~i~~
~~f~~
~~g~~
~~d~~
~~c~~





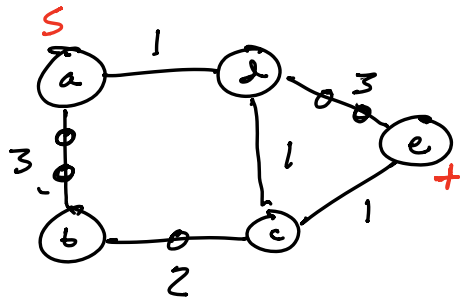
Graphs with weights:

Option 1 (Lazy programmer):

add vertices to edges with weight > 1
 $\# \text{ vertices} = w(e) - 1$

$$k = \max(w(e) \text{ for } e \in E)$$

$$O(n + mk)$$



Option 2 (Intuitive) $O(|E| \cdot |V|)$

Initialization: For every $v \in V$
 $\text{dist}(s, v) = \infty$

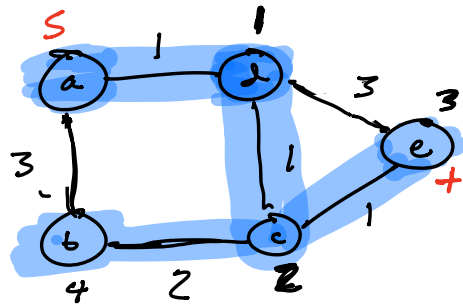
Initialize $X = \{s\}$

for $i = 2$ to $|V|$

Among nodes $V - X$ find node v that's the closest

Add v to X

Update $\text{dist}(s, v)$



d' dist

Option 3: Dijkstra's Algorithm

Initialization

For every $v \in V$ $\text{dist}(s, v) = d'(s, v) = \infty$

$X = \{s\}$

$d'(s, s) = 0$

Update $d'(s, u)$ for $u \in \text{Adj}(s)$

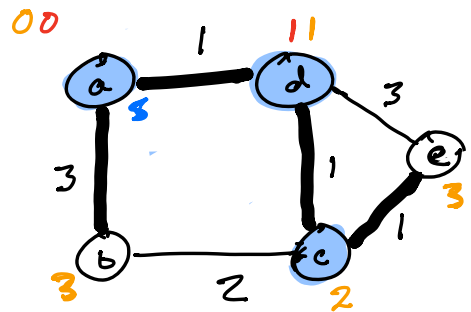
For $i = 1$ to $|V|$ do

Extract \rightarrow Let v realize $d'(s, v) = \min_{u \in V - X} d'(s, u)$
 Min function

$\text{dist}(s, v) = d'(s, v)$

$X = X \cup \{v\}$

Update $d'(s, u)$ for each u in $V - X$ Adj to v as

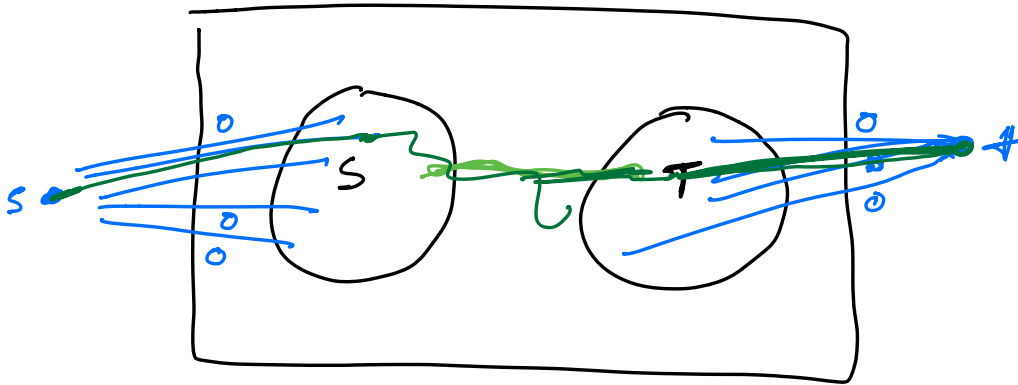


Update key $d'(s,u) = \min(d'(s,u), \text{dist}(s,v) + l(u,v))$
 $O(\text{deg}(v))$ $\text{deg}(v) - 1 = |E|$

Using Arrays $O(n \cdot n + m)$ $O(n^2 + m)$

Heap / Priority Queue Extract min $\log n$ decrease key $\log n$ $O((m+n) \log n)$

Fibonacci Heaps $\log n$ (Amortized) $O(n \log n + m)$



Dijkstra (s, G)

dist [t]

