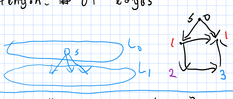
Lecture 20: short/shorter/shortest path

10/27/22

G: directed graph
s: source vertex
Compute shortest path in G from s
length = # of edges

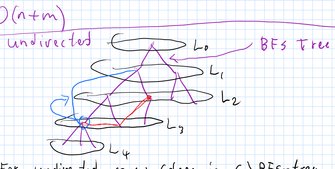


BFS with layers (distances)

mark all vertices as unvisited
mark s as visited
$L_0 \leftarrow \{s\}$
$i \leftarrow 0$
while L_i is not empty do
  $L_{i+1} \leftarrow \emptyset$
  for $x \in L_i$ do
    for $(x \to y) \in E(G)$ do // all neighbors of x
      if y is not visited then
        $L_{i+1} \leftarrow L_{i+1} \cup \{y\}$
        mark y as visited, and $d(y) \leftarrow i+1$.
        add $x \to y$ to the BFS tree
  $i \leftarrow i+1$

$O(n+m)$

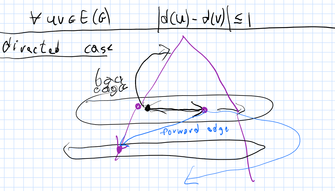
undirected



For undirected case: Edges in $G \setminus$ BFS-tree are between vertices of same layer, or between two consecutive levels.

$\forall uv \in E(G \setminus \quad |d(u) - d(v)| \le 1$

directed case

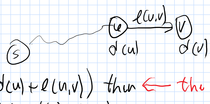


What to do if there are lengths on the edges?

positive lengths

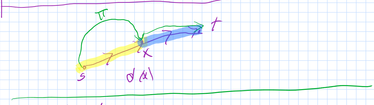
$\forall v \in V$ $d(v)$ candidate distance from s to v.

$d(s) = 0$ $\forall v \in V - s$ $d(v) = +\infty$



if $d(v) > (d(u) + \ell(u,v))$ then ← the edge $u \to v$ is tense
  $d(v) \leftarrow d(u) + \ell(u,v)$

Relax $(u \to v)$:
  if $d(v) > (d(u) + \ell(u \to v))$ then
    $d(v) \leftarrow d(u) + \ell(u \to v)$

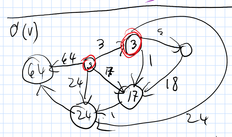


relax the edge $\xrightarrow{e}$ $\ell(e)$ being an integer

$\ell(e) = 4$ $\rightarrow\rightarrow\rightarrow\rightarrow$

$G \Rightarrow G'$ BFS(G')

$\xrightarrow{n^2}$



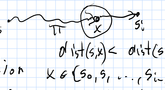
s_i: the ith closest vertex to s in G
$s_0 = s$
s_1

Dijkstra(G, s):
  $d(s) \leftarrow 0$, $\forall v \in V$ $d(v) \leftarrow +\infty$
  $H \leftarrow$ heap$((0,s), (+\infty, v_1), (+\infty, v_2), \ldots)$
  while H is not empty do
    $v \leftarrow$ extract-min(H)
    // $d(v)$ is now finalized
    for $(v \to x) \in E(G)$ do
      if $(d(v) + \ell(v \to x)) < d(x)$ do
        $d(x) \leftarrow d(v) + \ell(v \to x)$
        decreasekey$((d(x), x))$

heap: data-structure that stores pairs (v, key)
extract-min: return pair with min value in the heap
delete
insert
decreasekey$((v', k))$: updates $(v, k) \in H$ to be $H[v] \leftarrow v'$
replaces (v, k) by (v', k)
$v' < v$

Claim Dijkstra computes shortest paths distances correctly

Proof
Let s_i be the ith closest vertex to s in the "real" shortest distances.
Proof by induction on i.
$i = 0$: $s_0 = s$.
assume alg works correctly for $\forall i < \alpha$
Inductive step: prove that $d(s_\alpha)$ is computed correctly
Consider the shortest path π from to s to s_i



$dist(s,x) < dist(s, s_i)$

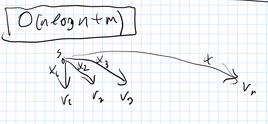
by induction $x \in \{s_0, s_1, \ldots, s_{i-1}\}$
it must be $d(x) = dist(s, x)$

$d(s_i) \le d(x) + \ell(x \to s_i) = dist(s,x) + \ell(x \to s_i)$
$= len(\pi) = dist(s, s_i)$
$\Downarrow$
$d(s_i)$
$\Rightarrow d(s_i) = dist(s, s_i)$. ∎

Running time

Heap implemented as balanced binary search trees
$O(\log n)$
$\Rightarrow O((n+m)\log n)$

Fibonacci heap

$O(\log n)$: insert/delete
$O(1)$: decrease key

$\boxed{O(n \log n + m)}$