

Heaps

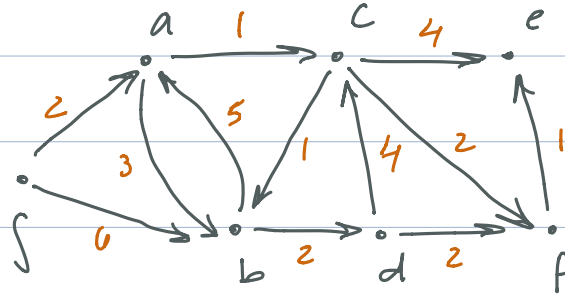
deletemin

→ insert

→ change

 $s: 0$ $s \rightarrow a: 2, b: 6$ $sa \rightarrow b: 5, c: 3$ $sac \rightarrow b: 4, c: 7, f: 5$ $sacb \rightarrow d: 6, c: 7, f: 5$

shortest path from s



for each v:

look at every neighbor of v

calculate distances

start w/ overestimate and * Only positive edges:
lower distance pred.

→ Algo. dist array, prev array, HEAP+1

 $H := \{s, 0\}$

Single-source shortest path

for $v \in V$: Initialization $O(|V|)$ Dijkstra's Algo $\text{dist}[v] = \infty, \text{prev}[v] = \text{nil}$ $\text{dist}[s] := 0$

Induction:

while $H \neq \emptyset$:

whenever you pull

 $v = \text{deletemin}(H)$ $O(|V|)$ deletemin
 $O(\log |V|)$

something off the heap

for $(v, w) \in E$:

is the shortest path

if $(\text{dist}[w] > \text{dist}[v] + \text{length}(v, w))$: $\text{dist}[w] = \text{dist}[v] + \text{length}(v, w)$ $\text{prev}[w] = v$ worst case $O(|E|)$ insert/change $\{w, \text{dist}[w] + 1\}$ insert/change
 $O(\log |V|)$ Runtime: $O((|E| + |V|) \cdot \log |V|) = O(|E| \cdot \log |V|)$

Runtime w/ linked list: $O(|V|^2)$

dist min: $O(|V|)$

insert/change: $O(1)$

Bellman-Ford works if there is a shortest path

update (v, w) : (no negative cycles)

if ($\text{dist}[w] > \text{dist}[v] + \text{length}(v, w)$):

$\text{dist}[w] := \text{dist}[v] + \text{length}(v, w)$

$\text{prev}[w] := v$

Algo

for $v \in V$:

$\text{dist}[v] := \infty$; $\text{prev}[v] := \text{nil}$

$\text{dist}[s] := 0$

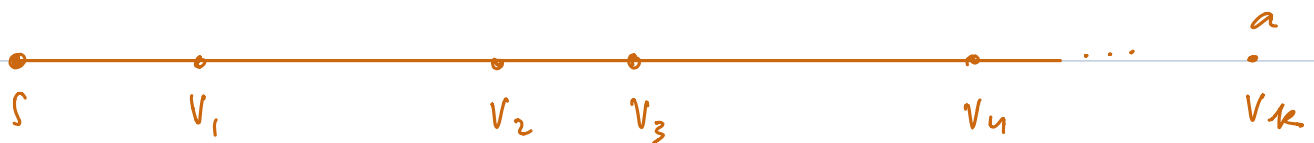
for $i = 1 \dots n-1$

$O(|V| \cdot |E|)$ worst case update for each edge

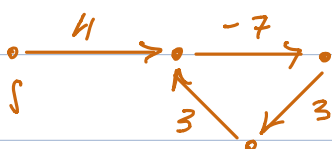
for $(v, w) \in E$:

update (v, w)

shortest path from s to $a = V_k$ is $s \rightarrow V_1 \rightarrow V_2 \rightarrow \dots \rightarrow V_k$



shortest path: negative cycle



invalid case