Lecture 27

Dynamic Programming: Bellman-Ford (contd.), Activity-Selection Problem

Recurrence for SSSP

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

We are interested in $dist(v, n-1)$ for $v \in V(G)$

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

If there is no path from s to v with at most i edges, then $dist(v, i) = \infty$

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

We would like to develop a recurrence so that $dist(v, i) = W$.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 1: One of the paths in S has at most $i - 1$ edges.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 1: One of the paths in S has at most $i - 1$ edges.

Case 2: All the paths in S have exactly i edges.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 1: One of the paths in S has at most $i - 1$ edges.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 1: One of the paths in S has at most $i - 1$ edges. Then,

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 1: One of the paths in S has at most $i - 1$ edges. Then,

$$W = dist(v, i - 1)$$

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 1**:** One of the paths in S has at most $i - 1$ edges. Then,

$$W = dist(v, i - 1)$$

↑

Clearly, $dist(v, i - 1) > W$ is not possible.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 1**:** One of the paths in S has at most $i - 1$ edges. Then,

$$W = dist(v, i - 1)$$

↑

$dist(v, i - 1) < W$ is also not possible.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 1**:** One of the paths in S has at most $i - 1$ edges. Then,

$$W = dist(v, i - 1)$$

$dist(v, i - 1) < W$ is also not possible. Otherwise, it will contradict.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 1**:** One of the paths in S has at most $i - 1$ edges. Then,

$$W = dist(v, i - 1) = dist(v, i)$$

$dist(v, i - 1) < W$ is also not possible. Otherwise, it will contradict.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 1: One of the paths in S has at most $i - 1$ edges. Then,

$$W = dist(v, i - 1) = dist(v, i)$$

$dist(v, i - 1) < W$ is also not possible. Otherwise, it will contradict.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 2: All the paths in S have exactly i edges.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

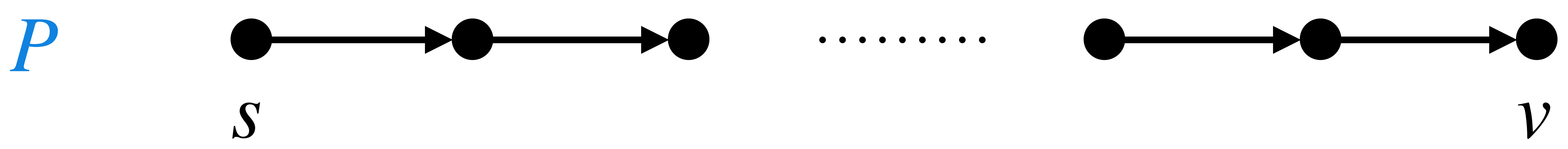
Case 2: All the paths in S have exactly i edges. Let P be one of the paths in S.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 2: All the paths in S have exactly i edges. Let P be one of the paths in S.



Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 2: All the paths in S have exactly i edges. Let P be one of the paths in S.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 2: All the paths in S have exactly i edges. Let P be one of the paths in S.

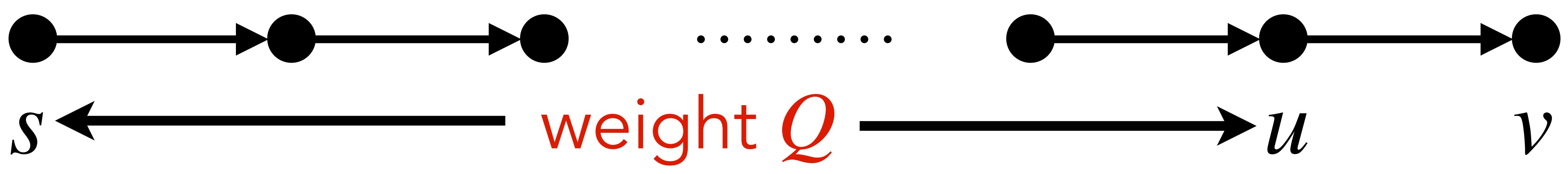
Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 2: All the paths in S have exactly i edges. Let P be one of the paths in S.

P



W

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 2: All the paths in S have exactly i edges. Let P be one of the paths in S.

$$W = Q + w(u, v)$$

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 2: All the paths in S have exactly i edges. Let P be one of the paths in S.

$$W = Q + w(u, v)$$

$$Q = dist(u, i-1)$$

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 2: All the paths in S have exactly i edges. Let P be one of the paths in S.

$$W = Q + w(u, v)$$

$Q = dist(u, i-1)$, $\because$ a shorter path from s to u with at most $i-1$ edges will contradict

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 2: All the paths in S have exactly i edges. Let P be one of the paths in S.

$$W = Q + w(u, v)$$

$Q = dist(u, i-1)$, $\because$ a shorter path from s to u with at most $i-1$ edges will contradict

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 2: All the paths in S have exactly i edges. Let P be one of the paths in S.

$$W = Q + w(u, v) = dist(u, i-1) + w(u, v)$$

$Q = dist(u, i-1)$, $\because$ a shorter path from s to u with at most $i-1$ edges will contradict

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 2: All the paths in S have exactly i edges. Let P be one of the paths in S.

$$W = Q + w(u, v) = dist(u, i-1) + w(u, v) = dist(v, i)$$

$Q = dist(u, i-1)$, $\because$ a shorter path from s to u with at most $i-1$ edges will contradict

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 2: All the paths in S have exactly i edges. Let P be one of the paths in S.

$$W = Q + w(u, v) = dist(u, i-1) + w(u, v) = dist(v, i)$$

$Q = dist(u, i-1)$, $\because$ a shorter path from s to u with at most $i-1$ edges will contradict

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Case 1: One of the paths in S has at most $i - 1$ edges. Then,

$$W = dist(v, i-1) = dist(v, i)$$

Case 2: All the paths in S have exactly i edges. Let P be one of the paths in S.

$$W = Q + w(u, v) = dist(u, i-1) + w(u, v) = dist(v, i)$$

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

$$dist(v, i) \ =$$

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

$$dist(v, i) = \left\{ \right.$$

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

$$dist(v, i) = \begin{cases} 0, & \text{if } i = 0 \text{ and } v = s \\ & \\ & \end{cases}$$

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

$$dist(v, i) = \begin{cases} 0, & \text{if } i = 0 \text{ and } v = s \\ \infty, & \text{if } i = 0 \text{ and } v \in V - \{s\} \\ & \\ & \end{cases}$$

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

$$dist(v, i) = \begin{cases} 0, & \text{if } i = 0 \text{ and } v = s \\ \infty, & \text{if } i = 0 \text{ and } v \in V - \{s\} \\ \quad\quad\quad\quad , & \text{otherwise} \end{cases}$$

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

$$dist(v, i) = \begin{cases} 0, & \text{if } i = 0 \text{ and } v = s \\ \infty, & \text{if } i = 0 \text{ and } v \in V - \{s\} \\ min \left\{ \qquad \right. , & \text{otherwise} \end{cases}$$

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

$$dist(v,i) = \begin{cases} 0, & \text{if } i = 0 \text{ and } v = s \\ \infty, & \text{if } i = 0 \text{ and } v \in V - \{s\} \\ min \begin{cases} dist(v, i-1) \\ \\ \end{cases} & \text{, otherwise} \end{cases}$$

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

$$dist(v, i) = \begin{cases} 0, & \text{if } i = 0 \text{ and } v = s \\ \infty, & \text{if } i = 0 \text{ and } v \in V - \{s\} \\ min \begin{cases} dist(v, i-1) \\ min_{(u,v) \in E}(d(u, i-1) + w(u, v)) \end{cases}, & \text{otherwise} \end{cases}$$

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

$$dist(v,i) = \begin{cases} 0, & \text{if } i = 0 \text{ and } v = s \\ \infty, & \text{if } i = 0 \text{ and } v \in V - \{s\} \\ min \begin{cases} dist(v, i-1) \\ min_{(u,v)\in E}(d(u, i-1) + w(u,v)) \end{cases}, & \text{otherwise} \end{cases}$$

Recurrence for SSSP

Let $dist(v, i)$ = weight of a shortest path among all the paths from s to v with at most i edges.

Let S be the set of all the shortest paths, each of weight W, from s to v with at most i edges.

$$dist(v, i) = \begin{cases} 0, & \text{if } i = 0 \text{ and } v = s \\ \infty, & \text{if } i = 0 \text{ and } v \in V - \{s\} \\ min \begin{cases} dist(v, i-1) \\ min_{(u,v)\in E}(d(u, i-1) + w(u, v)) \end{cases}, & \text{otherwise} \end{cases}$$

Why this will not be less than W?

Bellman-Ford: Bottom-Up DP for SSSP

Bellman-Ford: Bottom-Up DP for SSSP

SSSP(G, s)

Bellman-Ford: Bottom-Up DP for SSSP

SSSP(G, s)

1. $d[1:n],\ dist[1:n][0:n-1] = \{\infty, \infty, \ldots, \infty\}$

Bellman-Ford: Bottom-Up DP for SSSP

$n = |V(G)|$

SSSP(G, s)

1. $d[1:n],\ dist[1:n][0:n-1] = \{\infty, \infty, \ldots, \infty\}$

Bellman-Ford: Bottom-Up DP for SSSP

SSSP(G, s)

1. $d[1:n],\ dist[1:n][0:n-1] = \{\infty, \infty, \ldots, \infty\}$

$dist[v][i]$ stores weight of a shortest path from s to v using at most i edges.

Bellman-Ford: Bottom-Up DP for SSSP

$dist[v][i]$ stores weight of a shortest path from s to v using at most i edges.

SSSP(G, s)

1. $d[1:n],\ dist[1:n][0:n-1] = \{\infty, \infty, \ldots, \infty\}$
2. $dist[s][0] = 0$

Bellman-Ford: Bottom-Up DP for SSSP

SSSP(G, s)

1. $d[1:n],\ dist[1:n][0:n-1] = \{\infty, \infty, \ldots, \infty\}$
2. $dist[s][0] = 0$
3. **for** $i = 1$ **to** $n - 1$

$dist[v][i]$ stores weight of a shortest path from s to v using at most i edges.

Bellman-Ford: Bottom-Up DP for SSSP

SSSP(G, s)

1. $d[1:n],\ dist[1:n][0:n-1] = \{\infty, \infty, \ldots, \infty\}$
2. $dist[s][0] = 0$
3. **for** $i = 1$ **to** $n - 1$
4. **for** $v \in V(G)$

$dist[v][i]$ stores weight of a shortest path from s to v using at most i edges.

Bellman-Ford: Bottom-Up DP for SSSP

SSSP(G, s)

1. $d[1:n],\ dist[1:n][0:n-1] = \{\infty, \infty, \ldots, \infty\}$
2. $dist[s][0] = 0$
3. **for** $i = 1$ **to** $n - 1$
4.   **for** $v \in V(G)$
5. $dist[v][i] = dist[v][i-1]$

$dist[v][i]$ stores weight of a shortest path from s to v using at most i edges.

Bellman-Ford: Bottom-Up DP for SSSP

SSSP(G, s)

1. $d[1:n],\ dist[1:n][0:n-1] = \{\infty, \infty, \ldots, \infty\}$
2. $dist[s][0] = 0$
3. **for** $i = 1$ **to** $n - 1$
4. **for** $v \in V(G)$
5. $dist[v][i] = dist[v][i-1]$
6. **for** each edge (u, v)

$dist[v][i]$ stores weight of a shortest path from s to v using at most i edges.

Bellman-Ford: Bottom-Up DP for SSSP

$dist[v][i]$ stores weight of a shortest path from s to v using at most i edges.

SSSP(G, s)

1. $d[1:n],\ dist[1:n][0:n-1] = \{\infty, \infty, \ldots, \infty\}$
2. $dist[s][0] = 0$
3. **for** $i = 1$ **to** $n - 1$
4.   **for** $v \in V(G)$
5. $dist[v][i] = dist[v][i-1]$
6.   **for** each edge (u, v)
7. $dist[v][i] = min(dist[v][i], dist[u][i-1] + w(u, v))$

Bellman-Ford: Bottom-Up DP for SSSP

```
SSSP(G, s)
d[1 : n], dist[1 : n][0 : n − 1] = {∞, ∞, …, ∞}
dist[s][0] = 0
for i = 1 to n − 1
    for v ∈ V(G)
        dist[v][i] = dist[v][i − 1]
        for each edge (u, v)
            dist[v][i] = min(dist[v][i], dist[u][i − 1] + w(u, v))
for v ∈ V(G)
```

$dist[v][i]$ stores weight of a shortest path from s to v using at most i edges.

Bellman-Ford: Bottom-Up DP for SSSP

SSSP(G, s)

1. $d[1:n],\ dist[1:n][0:n-1] = \{\infty, \infty, \ldots, \infty\}$
2. $dist[s][0] = 0$
3. **for** $i = 1$ **to** $n - 1$
4.   **for** $v \in V(G)$
5. $dist[v][i] = dist[v][i-1]$
6.   **for** each edge (u, v)
7. $dist[v][i] = min(dist[v][i], dist[u][i-1] + w(u, v))$
8. **for** $v \in V(G)$
9. $d[v] = dist[v][n-1]$

$dist[v][i]$ stores weight of a shortest path from s to v using at most i edges.

Bellman-Ford: Bottom-Up DP for SSSP

SSSP(G, s)

```
d[1 : n], dist[1 : n][0 : n − 1] = {∞, ∞, …, ∞}
dist[s][0] = 0
for i = 1 to n − 1
    for v ∈ V(G)
        dist[v][i] = dist[v][i − 1]
            for each edge (u, v)
                dist[v][i] = min(dist[v][i], dist[u][i − 1] + w(u, v))
for v ∈ V(G)
    d[v] = dist[v][n − 1]
return d
```

$dist[v][i]$ stores weight of a shortest path from s to v using at most i edges.

Bellman-Ford: Bottom-Up DP for SSSP

```
SSSP(G, s)
d[1 : n], dist[1 : n][0 : n − 1] = {∞, ∞, …, ∞}
dist[s][0] = 0
for i = 1 to n − 1
    for v ∈ V(G)
        dist[v][i] = dist[v][i − 1]
            for each edge (u, v)
                dist[v][i] = min(dist[v][i], dist[u][i − 1] + w(u, v))
for v ∈ V(G)
    d[v] = dist[v][n − 1]
return d
```

Bellman-Ford: Bottom-Up DP for SSSP

SSSP(G, s)

$d[1:n],\ dist[1:n][0:n-1] = \{\infty, \infty, \ldots, \infty\}$
$dist[s][0] = 0$
for $i = 1$ **to** $n - 1$ ← $\Theta(|V|)$
  for $v \in V(G)$
    $dist[v][i] = dist[v][i-1]$
      for each edge (u, v)
        $dist[v][i] = min(dist[v][i], dist[u][i-1] + w(u, v))$
for $v \in V(G)$
  $d[v] = dist[v][n-1]$
return d

Bellman-Ford: Bottom-Up DP for SSSP

SSSP(G, s)

$d[1:n],\ dist[1:n][0:n-1] = \{\infty, \infty, \ldots, \infty\}$

$dist[s][0] = 0$

for $i = 1$ **to** $n - 1$ — $\Theta(|V|)$

 for $v \in V(G)$ — $\Theta(|V| + |E|)$

  $dist[v][i] = dist[v][i-1]$

  for each edge (u, v)

   $dist[v][i] = min(dist[v][i], dist[u][i-1] + w(u, v))$

for $v \in V(G)$

 $d[v] = dist[v][n-1]$

return d

Bellman-Ford: Bottom-Up DP for SSSP

SSSP(G, s)

$d[1:n],\ dist[1:n][0:n-1] = \{\infty, \infty, \ldots, \infty\}$
$dist[s][0] = 0$
for $i = 1$ **to** $n - 1$ — $\Theta(|V|)$
    for $v \in V(G)$ — $\Theta(|V| + |E|)$
        $dist[v][i] = dist[v][i-1]$
        for each edge (u, v) — $\Theta(|V| + |E|)$
            $dist[v][i] = min(dist[v][i], dist[u][i-1] + w(u, v))$
for $v \in V(G)$ — $\Theta(|V|)$
    $d[v] = dist[v][n-1]$
return d

Bellman-Ford: Bottom-Up DP for SSSP

```
SSSP(G, s)
d[1 : n], dist[1 : n][0 : n − 1] = {∞, ∞, …, ∞}
dist[s][0] = 0
for i = 1 to n − 1                      ← Θ(|V|)
    for v ∈ V(G)                        ← Θ(|V| + |E|)
        dist[v][i] = dist[v][i − 1]
        for each edge (u, v)            ← Θ(|V| + |E|)
            dist[v][i] = min(dist[v][i], dist[u][i − 1] + w(u, v))
for v ∈ V(G)                            ← Θ(|V|)
    d[v] = dist[v][n − 1]
return d
```

Time Complexity:

$\Theta(|V| . (|V| + |E|))$

Bellman-Ford: Bottom-Up DP for SSSP

SSSP(G, s)

$d[1:n]$, $dist[1:n][0:n-1] = \{\infty, \infty, \ldots, \infty\}$

$dist[s][0] = 0$

for $i = 1$ **to** $n - 1$

    for $v \in V(G)$

        $dist[v][i] = dist[v][i-1]$

        for each edge (u, v)

            $dist[v][i] = min(dist[v][i], dist[u][i-1] + w(u, v))$

for $v \in V(G)$

    $d[v] = dist[v][n-1]$

return d

Time Complexity:

$\Theta(|V| . (|V| + |E|))$

Bellman-Ford: Bottom-Up DP for SSSP

SSSP(G, s)

1. $d[1:n],\ dist[1:n][0:n-1] = \{\infty, \infty, \ldots, \infty\}$
2. $dist[s][0] = 0$
3. **for** $i = 1$ **to** $n - 1$
4.   **for** $v \in V(G)$
5. $dist[v][i] = dist[v][i-1]$
6.   **for** each edge (u, v)
7. $dist[v][i] = min(dist[v][i], dist[u][i-1] + w(u, v))$
8. **for** $v \in V(G)$
9. $d[v] = dist[v][n-1]$
10. **return** d

Modify line 4 to 7 so that this cost becomes only $\Theta(|E|)$.

Time Complexity:

$\Theta(|V| . (|V| + |E|))$

Activity-Selection

Activity-Selection

Act-Sel:

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i.

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output:

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Two activities a_i and a_j are mutually compatible if either $s_i \geq f_j$ or $s_j \geq f_i$.

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Assumption: Given activities are sorted in monotonically increasing order of finish time.

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Assumption: Given activities are sorted in monotonically increasing order of finish time.

Example:

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Assumption: Given activities are sorted in monotonically increasing order of finish time.

Example:

i	1	2	3	4	5	6	7	8	9	10	11
s_i											
f_i											

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Assumption: Given activities are sorted in monotonically increasing order of finish time.

Example:

i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	5	3	5	6	7	8	2	12
f_i	4	5	6	7	9	9	10	11	12	14	16

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Assumption: Given activities are sorted in monotonically increasing order of finish time.

Example:

i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	5	3	5	6	7	8	2	12
f_i	4	5	6	7	9	9	10	11	12	14	16

$\{a_1, a_4, a_8, a_{11}\}$ is a largest-size subset of mutually compatible activities.

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Assumption: Given activities are sorted in monotonically increasing order of finish time.

Example:

i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	5	3	5	6	7	8	2	12
f_i	4	5	6	7	9	9	10	11	12	14	16

$\{a_1, a_4, a_8, a_{11}\}$ is a largest-size subset of mutually compatible activities.