

COMP2521 24T3

Graphs (VI)

Dijkstra's Algorithm

Hao Xue

`cs2521@cse.unsw.edu.au`

shortest path
dijkstra's algorithm

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

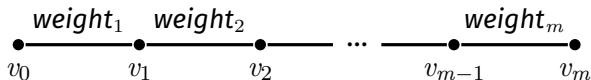
Other
Algorithms

Appendix

In a weighted graph...

A **path** is a sequence of edges
connected end-to-end

$(v_0, v_1, w_1), (v_1, v_2, w_2), \dots, (v_{m-1}, v_m, w_m)$



The **cost** of a path is
the sum of edge weights along the path

The **shortest path** between two vertices s and t is
the path from s to t with minimum cost

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

Variations on shortest path problem:

- **Source-target** shortest path
 - Shortest path from source vertex s to target vertex t
- **Single-source** shortest path
 - Shortest path from source vertex s to all other vertices
- **All-pairs** shortest path
 - Shortest path between all pairs of source and target vertices

Algorithm

Pseudocode

Example

Path Finding

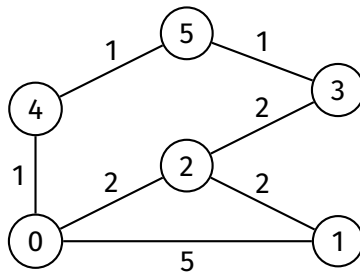
Vertex Set

Analysis

Other
Algorithms

Appendix

In a weighted graph,
a path with more edges may be “shorter” than
a path with fewer edges



Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

Invented by Dutch computer scientist
Edsger W. Dijkstra in 1956



Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

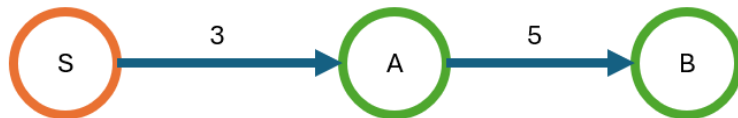
Other
Algorithms

Appendix

Dijkstra's algorithm
is used to find the **shortest path**
in a **weighted graph** with non-negative weights

Dijkstra's Algorithm Example

Algorithm
Pseudocode
Example
Path Finding
Vertex Set
Analysis
Other
Algorithms
Appendix





Distance starting S:

$$D(S, A) = 3$$

$$D(S, B) = \infty$$



Distance starting S:

$$D(S, A) = 3$$

$$D(S, B) = \infty$$

$$D(S, A) + D(A, B) = 8 < \infty$$

$$D(S, B) = 8 \text{ (via A)}$$

Algorithm

Edge relaxation

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

Data structures used in Dijkstra's algorithm:

- Distance array (`dist`)
 - To keep track of shortest currently known distance to each vertex
- Predecessor array (`pred`)
 - Same purpose as in BFS/DFS
 - To keep track of the predecessor of each vertex on the shortest currently known path to that vertex
 - Used to construct the shortest path
- Set of vertices
 - Stores unexplored vertices

Algorithm

Edge relaxation

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

- ① Create and initialise data structures
 - Create distance array, initialised to infinity
 - In C, can use `INT_MAX` (from `<limits.h>`)
 - Create predecessor array, initialised to -1
 - Initialise set of vertices to contain all vertices
- ② Set distance of source vertex (s) to 0
- ③ While set of vertices is not empty:
 - ① Remove vertex from vertex set with smallest distance in distance array
 - Let this vertex be v
 - ② **Explore** v - that is, for each edge $v - w$:
 - Check if using this edge gives a shorter path to w
 - If so, update w 's distance and predecessor - this is called **edge relaxation**

Algorithm

Edge relaxation

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

During Dijkstra's algorithm, the `dist` and `pred` arrays:

- contain data about the shortest path discovered *so far*
- need to be updated if a shorter path to some vertex is found
 - this is done via **edge relaxation**

Algorithm

Edge relaxation

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

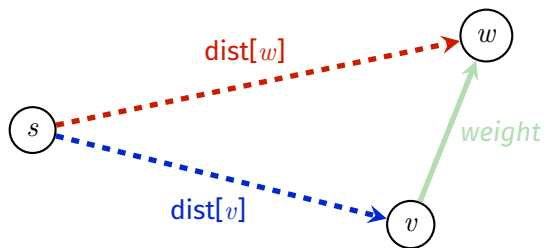
Suppose we are considering edge (v, w, weight) .



Suppose we are considering edge (v, w, weight) .

We have the following data:

- $\text{dist}[v]$ - length of shortest known path from s to v
- $\text{dist}[w]$ - length of shortest known path from s to w (which may be ∞)



Algorithm

Edge relaxation

Pseudocode

Example

Path Finding

Vertex Set

Analysis

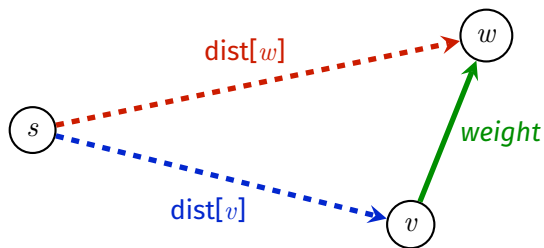
Other
Algorithms

Appendix

Suppose we are considering edge (v, w, weight) .

We have the following data:

- $\text{dist}[v]$ - length of shortest known path from s to v
- $\text{dist}[w]$ - length of shortest known path from s to w (which may be ∞)



In edge relaxation, we take the shortest known path from s to v and *extend* it using edge (v, w, weight) to create a *new* path from s to w .

Algorithm

Edge relaxation

Pseudocode

Example

Path Finding

Vertex Set

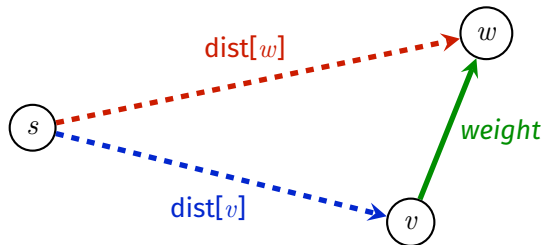
Analysis

Other
Algorithms

Appendix

Now we have two paths from s to w :

- Shortest known path
- New path via v



If the new path is shorter, then we update $\text{dist}[w]$ and $\text{pred}[w]$.

```
if  $\text{dist}[v] + \text{weight} < \text{dist}[w]$ :  
     $\text{dist}[w] = \text{dist}[v] + \text{weight}$   
     $\text{pred}[w] = v$ 
```


Algorithm

Edge relaxation

Pseudocode

Example

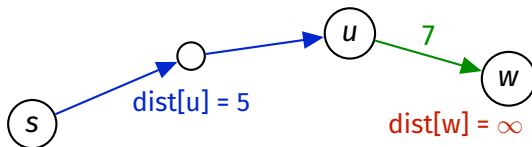
Path Finding

Vertex Set

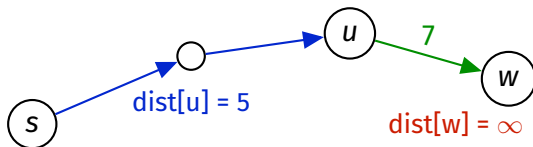
Analysis

Other
Algorithms

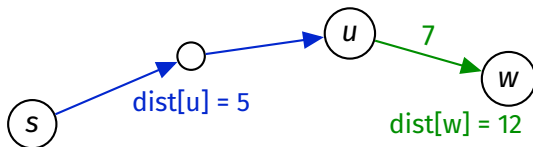
Appendix

Before relaxation along $(u, w, 7)$ 

	...	$[u]$...	$[w]$
dist	...	5	...	∞
pred	-1

Before relaxation along $(u, w, 7)$ 

	...	$[u]$...	$[w]$
dist	...	5	...	∞
pred	-1

After relaxation along $(u, w, 7)$ 

	...	$[u]$...	$[w]$
dist	...	5	...	12
pred	u

Algorithm

Edge relaxation

Pseudocode

Example

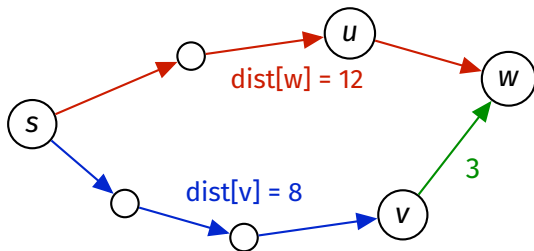
Path Finding

Vertex Set

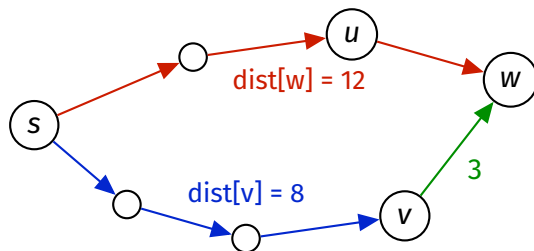
Analysis

Other
Algorithms

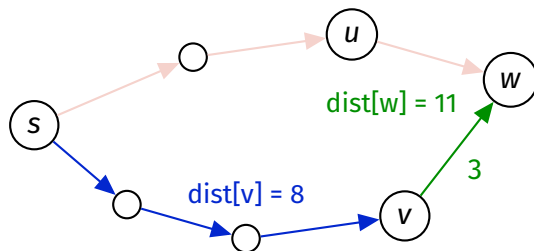
Appendix

Before relaxation along $(v, w, 3)$ 

	...	[u]	[v]	[w]
dist	...	5	8	12
pred	u

Before relaxation along $(v, w, 3)$ 

	...	$[u]$	$[v]$	$[w]$
dist	...	5	8	12
pred	u

After relaxation along $(v, w, 3)$ 

	...	$[u]$	$[v]$	$[w]$
dist	...	5	8	11
pred	v

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

`dijkstraSSSP(G , src):``Input: graph G , source vertex src` `create dist array, initialised to ∞` `create pred array, initialised to -1``create vSet containing all vertices of G` `$dist[src] = 0$` `while vSet is not empty:``find vertex v in vSet such that $dist[v]$ is minimal``remove v from vSet``for each edge (v, w, weight) in G :``relax along (v, w, weight)`

Algorithm

Pseudocode

Example

Path Finding

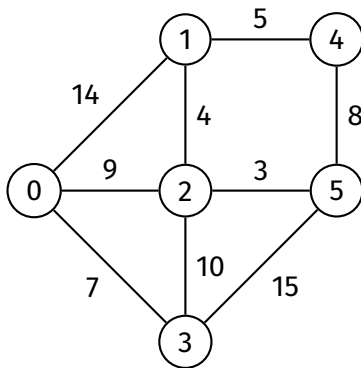
Vertex Set

Analysis

Other
Algorithms

Appendix

Dijkstra's algorithm starting at 0



Algorithm

Pseudocode

Example

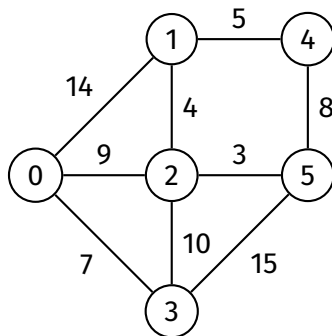
Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

Initialisation $\rightarrow vSet = \{0, 1, 2, 3, 4, 5\}$ 

while $vSet$ is not empty:
 find vertex v in $vSet$ such that
 $dist[v]$ is minimal
 and remove it from $vSet$

for each edge $(v, w, weight)$ in G :
 relax along $(v, w, weight)$

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	∞	∞	∞	∞	∞
pred	-1	-1	-1	-1	-1	-1

Algorithm

Pseudocode

Example

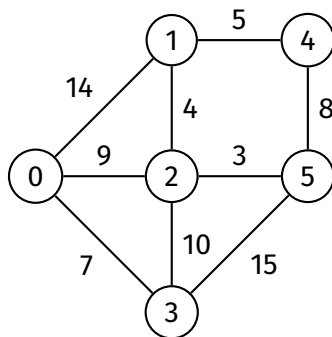
Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

After first iteration ($v = 0$) $\rightarrow vSet = \{1, 2, 3, 4, 5\}$ 

```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
    and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	∞
pred	-1	0	0	0	-1	-1

Algorithm

Pseudocode

Example

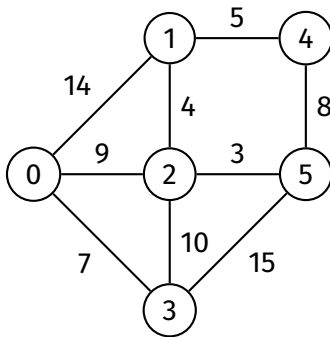
Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

After second iteration ($v = 3$) $\rightarrow vSet = \{1, 2, 4, 5\}$ 

```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
    and remove it from vSet
```

```
for each edge ( $v, w, \text{weight}$ ) in  $G$ :  
    relax along ( $v, w, \text{weight}$ )
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	22
pred	-1	0	0	0	-1	3

Algorithm

Pseudocode

Example

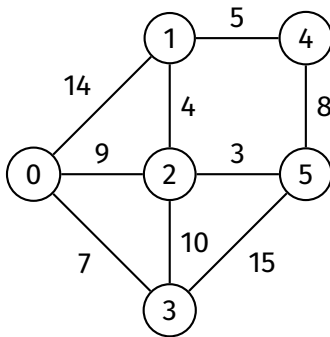
Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

After third iteration ($v = 2$) $\rightarrow vSet = \{1, 4, 5\}$ 

```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
    and remove it from vSet
```

```
for each edge ( $v, w, \text{weight}$ ) in  $G$ :  
    relax along ( $v, w, \text{weight}$ )
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	12
pred	-1	2	0	0	-1	2

Algorithm

Pseudocode

Example

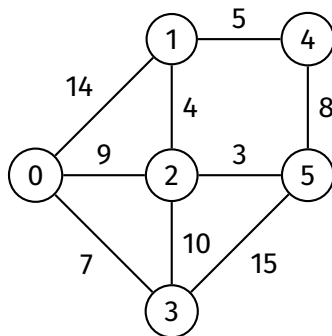
Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

After fourth iteration ($v = 5$) $\rightarrow vSet = \{1, 4\}$ 

```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
    and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	20	12
pred	-1	2	0	0	5	2

Algorithm

Pseudocode

Example

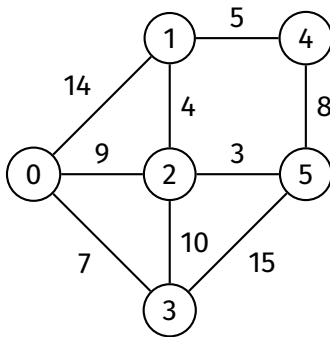
Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

After fifth iteration ($v = 1$) $\rightarrow vSet = \{4\}$ 

```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
        and remove it from vSet
```

```
for each edge ( $v, w, \text{weight}$ ) in  $G$ :  
    relax along ( $v, w, \text{weight}$ )
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	18	12
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

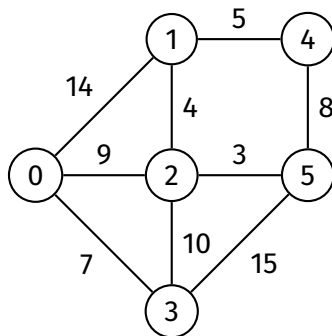
Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

After sixth iteration ($v = 4$) $vSet = \{ \}$ 

while $vSet$ is not empty:
 find vertex v in $vSet$ such that
 $dist[v]$ is minimal
 and remove it from $vSet$

for each edge $(v, w, weight)$ in G :
 relax along $(v, w, weight)$

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	18	12
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

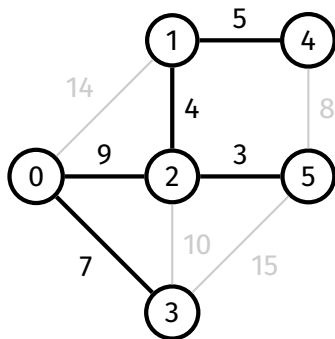
Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix



Done

```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
    and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	18	12
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

Path Finding

Example

Vertex Set

Analysis

Other
Algorithms

Appendix

The shortest path from the source vertex to any other vertex
can be constructed by tracing backwards through the predecessor array
(like for BFS)

Algorithm

Pseudocode

Example

Path Finding

Example

Vertex Set

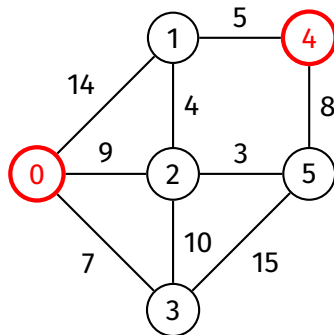
Analysis

Other
Algorithms

Appendix

Example: Shortest path from 0 to 4

0 4



	[0]	[1]	[2]	[3]	[4]	[5]
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

Path Finding

Example

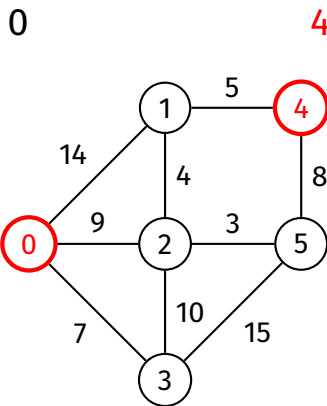
Vertex Set

Analysis

Other
Algorithms

Appendix

Example: Shortest path from 0 to 4



	[0]	[1]	[2]	[3]	[4]	[5]
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

Path Finding

Example

Vertex Set

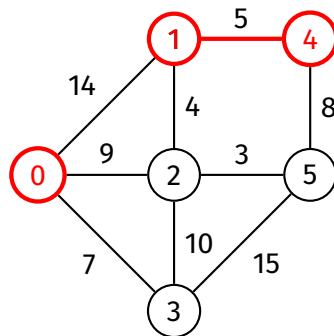
Analysis

Other
Algorithms

Appendix

Example: Shortest path from 0 to 4

0 1 → 4



	[0]	[1]	[2]	[3]	[4]	[5]
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

Path Finding

Example

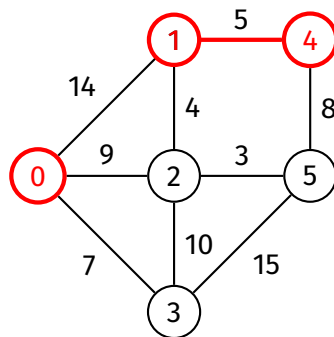
Vertex Set

Analysis

Other
Algorithms

Appendix

Example: Shortest path from 0 to 4

0 1 → 4

	[0]	[1]	[2]	[3]	[4]	[5]
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

Path Finding

Example

Vertex Set

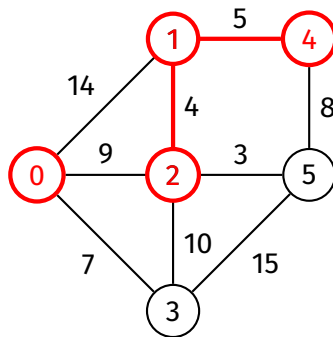
Analysis

Other
Algorithms

Appendix

Example: Shortest path from 0 to 4

0 2 → 1 → 4



	[0]	[1]	[2]	[3]	[4]	[5]
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

Path Finding

Example

Vertex Set

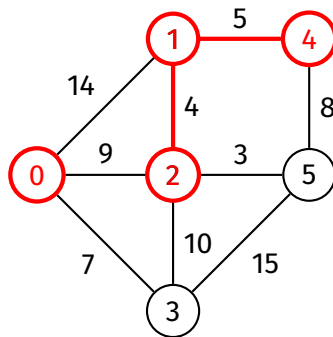
Analysis

Other
Algorithms

Appendix

Example: Shortest path from 0 to 4

0 2 → 1 → 4



	[0]	[1]	[2]	[3]	[4]	[5]
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

Path Finding

Example

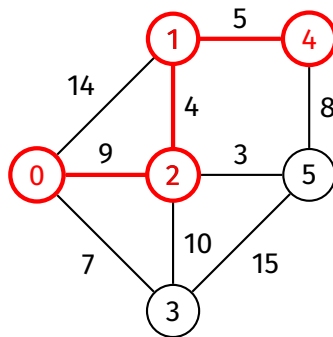
Vertex Set

Analysis

Other
Algorithms

Appendix

Example: Shortest path from 0 to 4

 $0 \longrightarrow 2 \longrightarrow 1 \longrightarrow 4$ 

	[0]	[1]	[2]	[3]	[4]	[5]
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

Path Finding

Example

Vertex Set

Analysis

Other
Algorithms

Appendix

How to find shortest path between two other vertices
(neither of which are the source vertex)?

Generally, you will need to rerun Dijkstra's algorithm from one of these vertices.

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

The vSet can be implemented in different ways:

- 1 Visited array
- 2 Explicit array/list of vertices
- 3 Priority queue

Visited array implementation:

- Similar to visited array in BFS/DFS
- Array of V booleans, initialised to false
- After exploring vertex v , set $\text{visited}[v]$ to true
- At the start of each iteration, find vertex v such that $\text{visited}[v]$ is false and $\text{dist}[v]$ is minimal $\Rightarrow O(V)$

Array/list of vertices implementation:

- Store all vertices in an array/linked list
- After exploring vertex v , remove v from array/linked list
- At the start of each iteration, find vertex in array/list such that $\text{dist}[v]$ is minimal $\Rightarrow O(V)$

Priority queue implementation:

- A priority queue is an ADT...
 - where each item has a priority
 - with two main operations:
 - **Insert:** insert item with priority
 - **Delete:** remove item with highest priority
- Use priority queue to store vertices, use *distance* to vertex as priority (smaller distance = higher priority)
- A good priority queue implementation has $O(\log n)$ insert and delete

Priority queues will be discussed in Week 9.

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Correctness

Time complexity

Other
Algorithms

Appendix

Proof by induction.

Aim is to prove that before and after each iteration:

- ❶ For all explored nodes s , $\text{dist}[s]$ is shortest distance from source to s
- ❷ For all unexplored nodes t , $\text{dist}[t]$ is shortest distance from source to t *via explored nodes only*

Ultimately, all nodes are explored, so by ❶:

- For all nodes v , $\text{dist}[v]$ is the shortest distance from source to v

Base case:

- Start of first iteration
 - ① holds, as there are no explored nodes
 - ② holds, because
 - $\text{dist}[\text{source}] = 0$
 - For all other nodes t , $\text{dist}[t] = \infty$

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Correctness

Time complexity

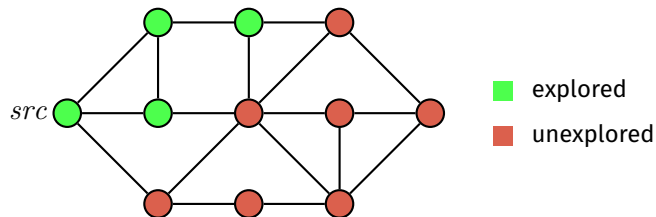
Other

Algorithms

Appendix

Induction step:

- Assume that ① and ② hold at the start of an iteration



Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Correctness

Time complexity

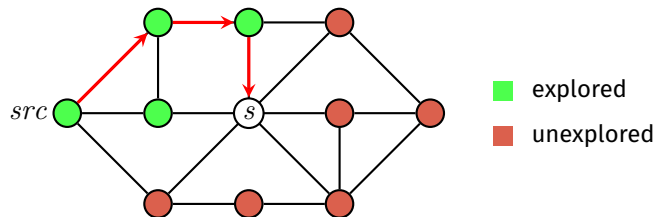
Other

Algorithms

Appendix

Induction step:

- Assume that ① and ② hold at the start of an iteration
- Let s be an unexplored node with minimum distance



Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Correctness

Time complexity

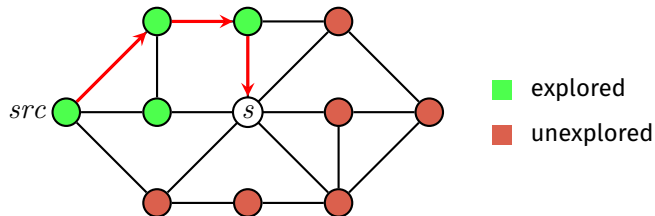
Other

Algorithms

Appendix

Induction step:

- Assume that ① and ② hold at the start of an iteration
- Let s be an unexplored node with minimum distance
- We claim that $\text{dist}[s]$ is the shortest distance from source to s



Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Correctness

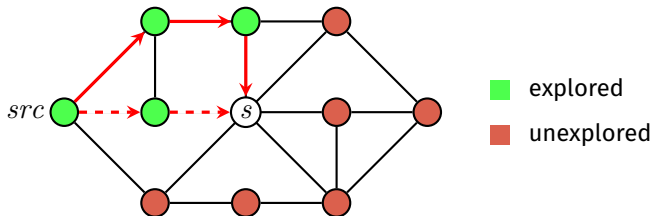
Time complexity

Other
Algorithms

Appendix

Induction step:

- Assume that ① and ② hold at the start of an iteration
- Let s be an unexplored node with minimum distance
- We claim that $\text{dist}[s]$ is the shortest distance from source to s
 - If there is a shorter path to s via explored nodes only, then $\text{dist}[s]$ would have been updated when exploring the predecessor of s on that path



Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Correctness

Time complexity

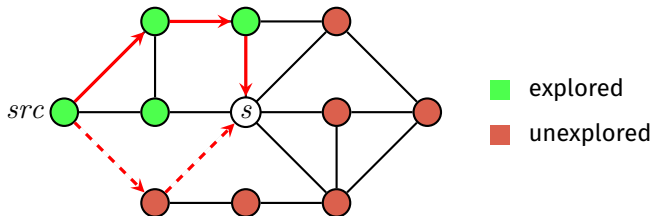
Other

Algorithms

Appendix

Induction step:

- Assume that ① and ② hold at the start of an iteration
- Let s be an unexplored node with minimum distance
- We claim that $\text{dist}[s]$ is the shortest distance from source to s
 - If there is a shorter path to s via explored nodes only, then $\text{dist}[s]$ would have been updated when exploring the predecessor of s on that path
 - If there is a shorter path to s via an unexplored node u , then $\text{dist}[u] < \text{dist}[s]$, which is a contradiction, since s has minimum distance out of all unexplored nodes. (non-negative weights!)



Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Correctness

Time complexity

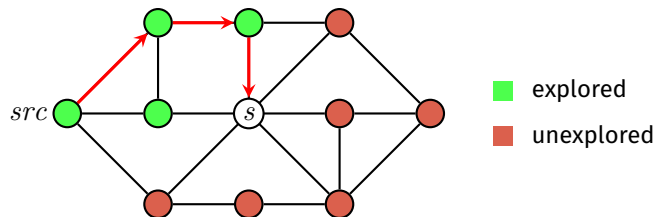
Other

Algorithms

Appendix

Induction step (continued):

- $\text{dist}[s]$ is the shortest distance from source to s



Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Correctness

Time complexity

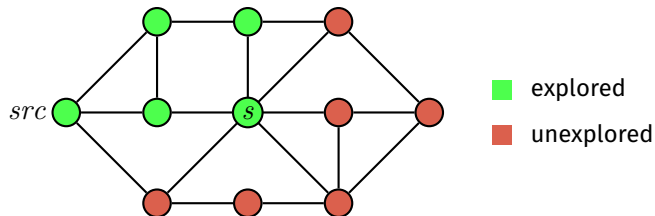
Other

Algorithms

Appendix

Induction step (continued):

- $\text{dist}[s]$ is the shortest distance from source to s
- After exploring s :



Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Correctness

Time complexity

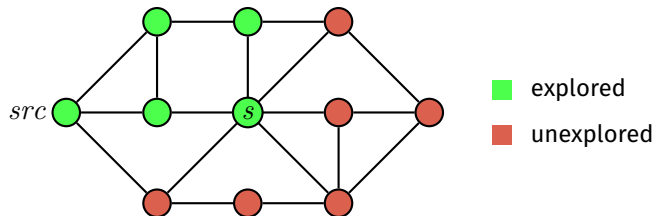
Other

Algorithms

Appendix

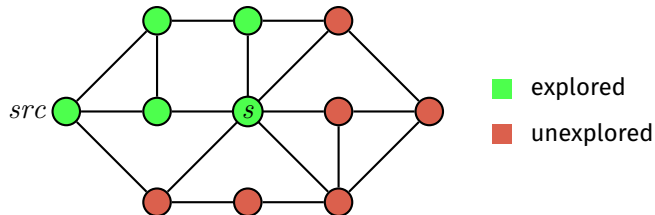
Induction step (continued):

- $\text{dist}[s]$ is the shortest distance from source to s
- After exploring s :
 - ① still holds for s , since $\text{dist}[s]$ is not updated while exploring s
 - Same for all other explored nodes



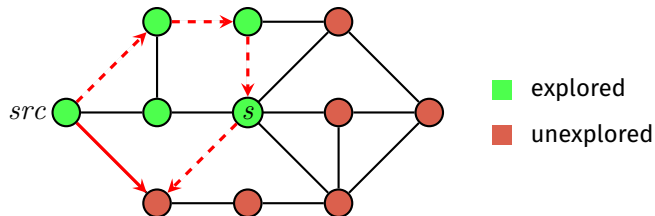
Induction step (continued):

- $\text{dist}[s]$ is the shortest distance from source to s
- After exploring s :
 - ① still holds for s , since $\text{dist}[s]$ is not updated while exploring s
 - Same for all other explored nodes
 - ② still holds for all unexplored nodes t , since:



Induction step (continued):

- $\text{dist}[s]$ is the shortest distance from source to s
- After exploring s :
 - ① still holds for s , since $\text{dist}[s]$ is not updated while exploring s
 - Same for all other explored nodes
 - ② still holds for all unexplored nodes t , since:
 - If there is a shorter path to t via s then we would have updated $\text{dist}[t]$ while exploring s



Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Correctness

Time complexity

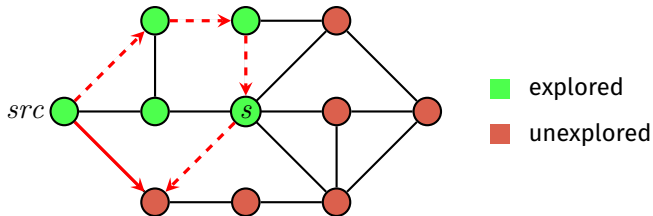
Other

Algorithms

Appendix

Induction step (continued):

- $\text{dist}[s]$ is the shortest distance from source to s
- After exploring s :
 - ① still holds for s , since $\text{dist}[s]$ is not updated while exploring s
 - Same for all other explored nodes
 - ② still holds for all unexplored nodes t , since:
 - If there is a shorter path to t via s then we would have updated $\text{dist}[t]$ while exploring s
 - Otherwise, we would not have updated $\text{dist}[t]$ and it would remain as it is



Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Correctness

Time complexity

Other

Algorithms

Appendix

Analysis:

- Each edge is considered once $\Rightarrow O(E)$
 - Undirected edges are considered once in each direction
- Outer loop has V iterations
- Every iteration, algorithm must find vertex v in $vSet$ with minimum distance - time complexity depends on $vSet$ implementation
 - Boolean array $\Rightarrow O(V)$ per iteration
 \Rightarrow overall cost = $O(E + V^2) = O(V^2)$
 - Array/list of vertices $\Rightarrow O(V)$ per iteration
 \Rightarrow overall cost = $O(E + V^2) = O(V^2)$
 - Priority queue $\Rightarrow O(\log V)$ per iteration
 \Rightarrow overall cost = $O(E + V \log V)$

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

Other
Algorithms

Appendix

- Floyd-Warshall Algorithm
 - All-pairs shortest path
 - Works for graphs with negative weights
- Bellman-Ford Algorithm
 - Single-source shortest path
 - Works for graphs with negative weights
 - Can detect negative cycles

Algorithm

Pseudocode

Example

Path Finding

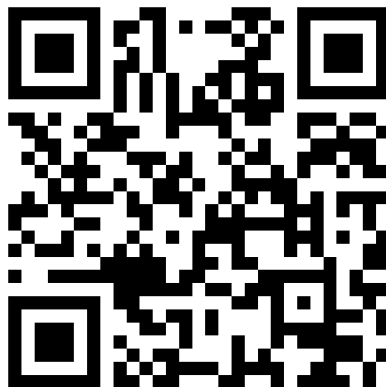
Vertex Set

Analysis

Other
Algorithms

Appendix

<https://forms.office.com/r/zEqxUXvmLR>



Algorithm

Pseudocode

Example

Path Finding

Vertex Set

Analysis

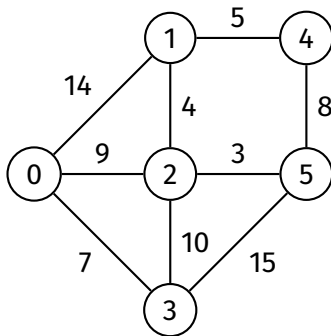
Other
Algorithms

Appendix

Example

Appendix

Initialisation



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
    and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	∞	∞	∞	∞	∞
pred	-1	-1	-1	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

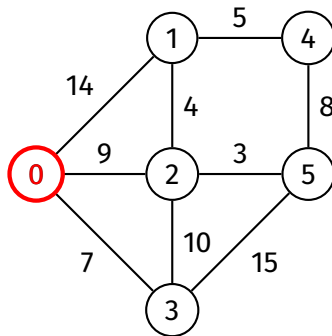
Analysis

Other
Algorithms

Appendix

Example

Remove 0 from vSet



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	∞	∞	∞	∞	∞
pred	-1	-1	-1	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

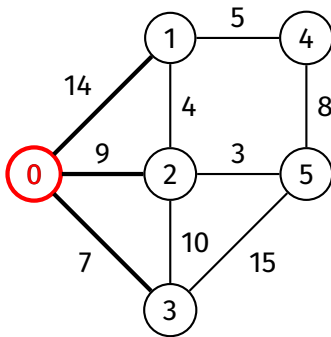
Analysis

Other
Algorithms

Appendix

Example

Explore 0



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
    and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	∞	∞	∞	∞	∞
pred	-1	-1	-1	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

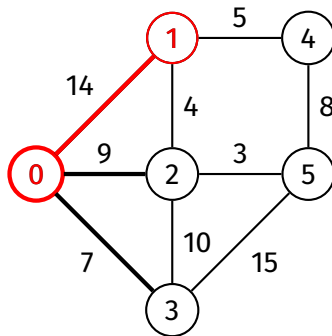
Analysis

Other
Algorithms

Appendix

Example

Relax along (0, 1, 14)



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
    and remove it from vSet
```

```
for each edge ( $v, w, \text{weight}$ ) in  $G$ :  
    relax along ( $v, w, \text{weight}$ )
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	∞	∞	∞	∞	∞
pred	-1	-1	-1	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

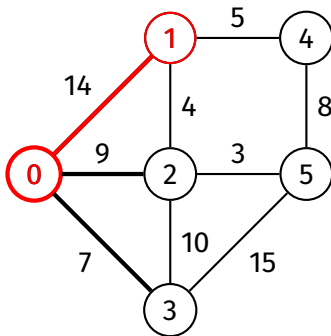
Analysis

Other
Algorithms

Appendix

Example

Relax along (0, 1, 14)
 $\text{dist}[0] + 14$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	∞	∞	∞	∞	∞
pred	-1	-1	-1	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

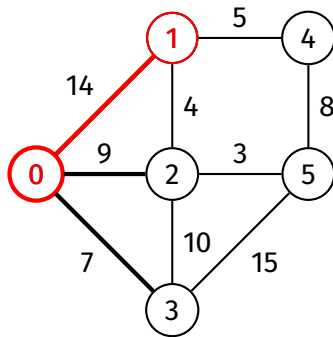
Analysis

Other
Algorithms

Appendix

Example

Relax along (0, 1, 14)
 $\text{dist}[0] + 14 = 14$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	∞	∞	∞	∞	∞
pred	-1	-1	-1	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

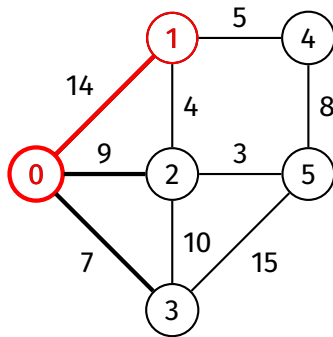
Analysis

Other
Algorithms

Appendix

Example

Relax along (0, 1, 14)
 $\text{dist}[0] + 14 = 14 < \text{dist}[1]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	∞	∞	∞	∞	∞
pred	-1	-1	-1	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

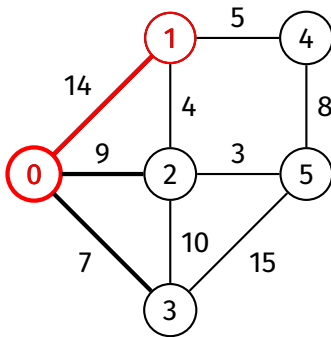
Analysis

Other
Algorithms

Appendix

Example

Relax along (0, 1, 14)
 $\text{dist}[0] + 14 = 14 < \text{dist}[1]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	∞	∞	∞	∞
pred	-1	0	-1	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

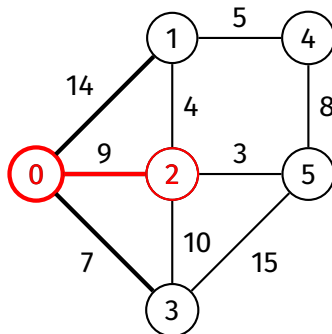
Analysis

Other
Algorithms

Appendix

Example

Relax along (0, 2, 9)



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
        and remove it from vSet
```

```
for each edge ( $v, w, \text{weight}$ ) in  $G$ :  
    relax along ( $v, w, \text{weight}$ )
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	∞	∞	∞	∞
pred	-1	0	-1	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

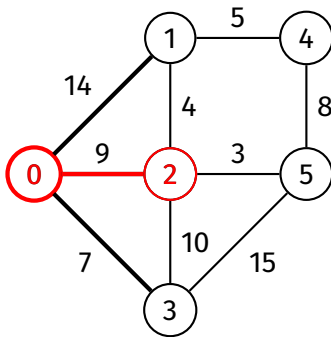
Analysis

Other
Algorithms

Appendix

Example

Relax along (0, 2, 9)
 $\text{dist}[0] + 9$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	∞	∞	∞	∞
pred	-1	0	-1	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

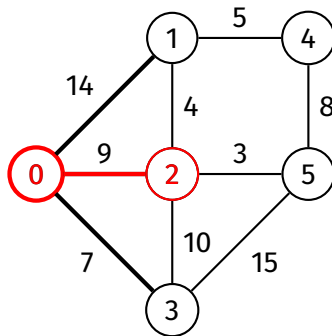
Analysis

Other
Algorithms

Appendix

Example

Relax along (0, 2, 9)
 $\text{dist}[0] + 9 = 9$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	∞	∞	∞	∞
pred	-1	0	-1	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

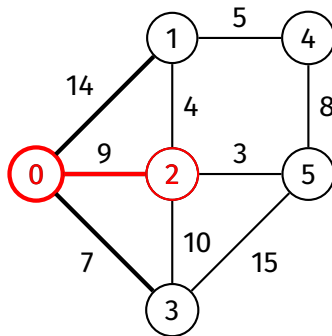
Analysis

Other
Algorithms

Appendix

Example

Relax along (0, 2, 9)
 $\text{dist}[0] + 9 = 9 < \text{dist}[2]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	∞	∞	∞	∞
pred	-1	0	-1	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

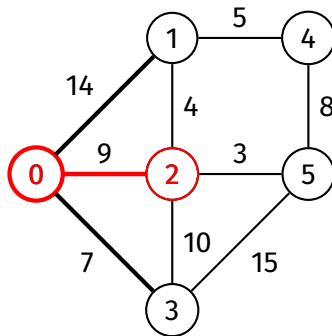
Analysis

Other
Algorithms

Appendix

Example

Relax along (0, 2, 9)
 $\text{dist}[0] + 9 = 9 < \text{dist}[2]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	∞	∞	∞
pred	-1	0	0	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

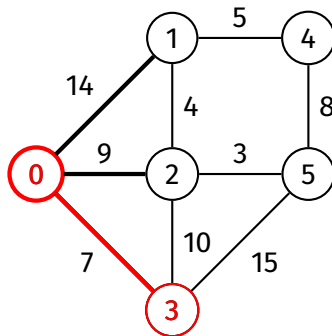
Analysis

Other
Algorithms

Appendix

Example

Relax along (0, 3, 7)



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
    and remove it from vSet
```

```
for each edge ( $v, w, \text{weight}$ ) in  $G$ :  
    relax along ( $v, w, \text{weight}$ )
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	∞	∞	∞
pred	-1	0	0	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

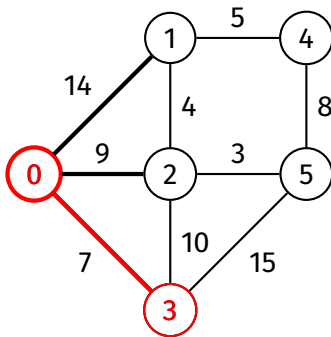
Analysis

Other
Algorithms

Appendix

Example

Relax along (0, 3, 7)
 $\text{dist}[0] + 7$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	∞	∞	∞
pred	-1	0	0	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

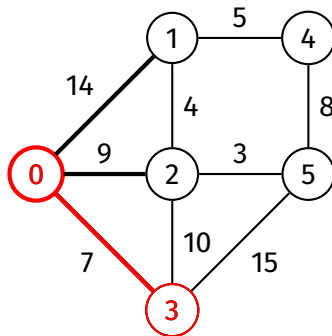
Analysis

Other
Algorithms

Appendix

Example

Relax along (0, 3, 7)
 $\text{dist}[0] + 7 = 7$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	∞	∞	∞
pred	-1	0	0	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

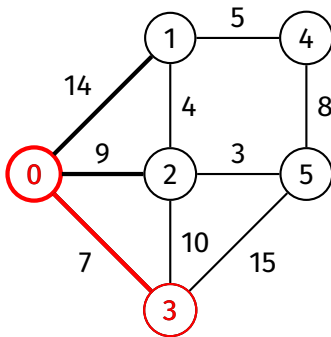
Analysis

Other
Algorithms

Appendix

Example

Relax along (0, 3, 7)
 $\text{dist}[0] + 7 = 7 < \text{dist}[3]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	∞	∞	∞
pred	-1	0	0	-1	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

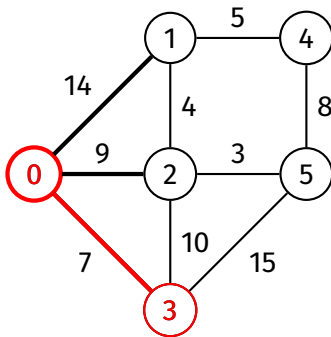
Analysis

Other
Algorithms

Appendix

Example

Relax along (0, 3, 7)
 $\text{dist}[0] + 7 = 7 < \text{dist}[3]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	∞
pred	-1	0	0	0	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

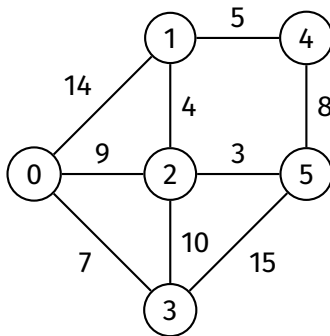
Analysis

Other
Algorithms

Appendix

Example

Done with exploring 0



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
    and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	∞
pred	-1	0	0	0	-1	-1

Algorithm

Pseudocode

Example

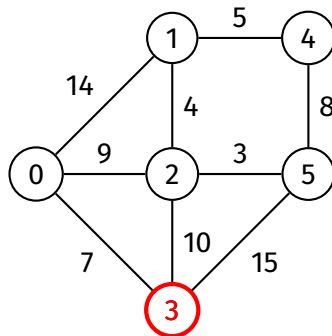
Path Finding

Vertex Set

Analysis

Other
AlgorithmsAppendix
Example

Remove 3 from vSet



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
    and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	∞
pred	-1	0	0	0	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

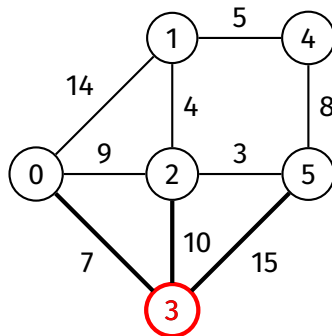
Analysis

Other
Algorithms

Appendix

Example

Explore 3



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	∞
pred	-1	0	0	0	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

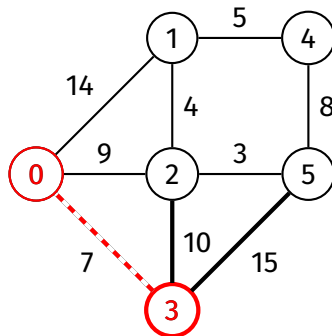
Analysis

Other
Algorithms

Appendix

Example

No need to consider (3, 0, 7)
(0 has already been explored)



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
        and remove it from vSet
```

```
for each edge ( $v, w, \text{weight}$ ) in  $G$ :  
    relax along ( $v, w, \text{weight}$ )
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	∞
pred	-1	0	0	0	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

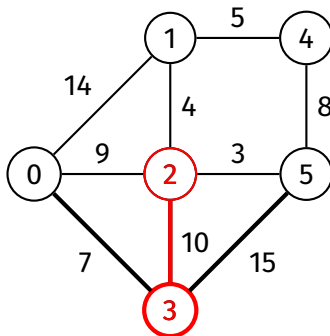
Analysis

Other
Algorithms

Appendix

Example

Relax along (3, 2, 10)



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
        and remove it from vSet
```

```
for each edge ( $v, w, \text{weight}$ ) in  $G$ :  
    relax along ( $v, w, \text{weight}$ )
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	∞
pred	-1	0	0	0	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

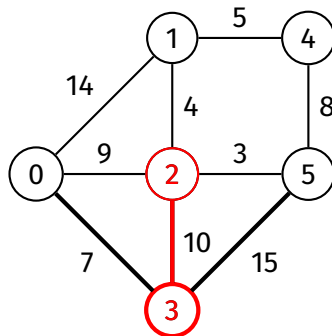
Analysis

Other
Algorithms

Appendix

Example

Relax along (3, 2, 10)
 $\text{dist}[3] + 10$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	∞
pred	-1	0	0	0	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

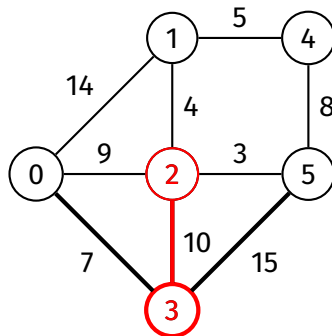
Analysis

Other
Algorithms

Appendix

Example

Relax along (3, 2, 10)
 $\text{dist}[3] + 10 = 17$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	∞
pred	-1	0	0	0	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

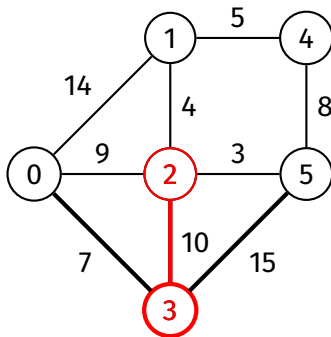
Analysis

Other
Algorithms

Appendix

Example

Relax along (3, 2, 10)
 $\text{dist}[3] + 10 = 17 \not< \text{dist}[2]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	∞
pred	-1	0	0	0	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

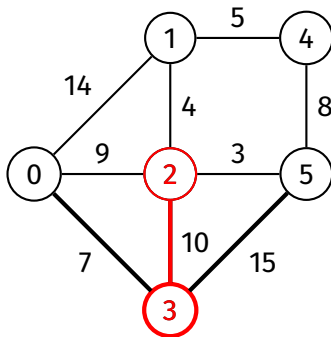
Analysis

Other
Algorithms

Appendix

Example

Relax along (3, 2, 10)
 $\text{dist}[3] + 10 = 17 \not< \text{dist}[2]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	∞
pred	-1	0	0	0	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

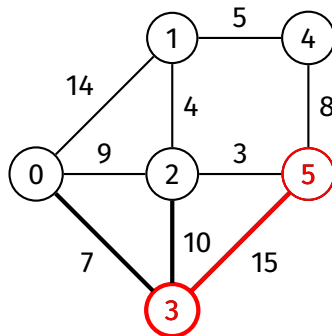
Analysis

Other
Algorithms

Appendix

Example

Relax along (3, 5, 15)



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	∞
pred	-1	0	0	0	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

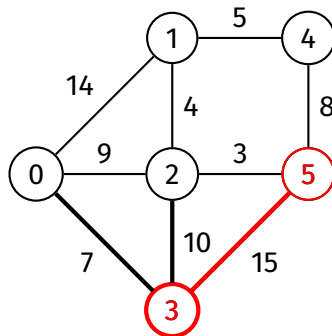
Analysis

Other
Algorithms

Appendix

Example

Relax along (3, 5, 15)
 $\text{dist}[3] + 15$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	∞
pred	-1	0	0	0	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

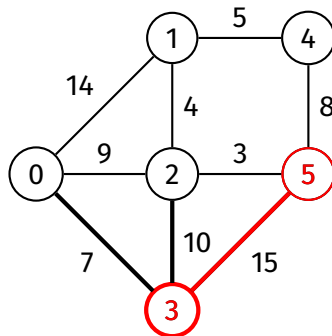
Analysis

Other
Algorithms

Appendix

Example

Relax along (3, 5, 15)
 $\text{dist}[3] + 15 = 22$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	∞
pred	-1	0	0	0	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

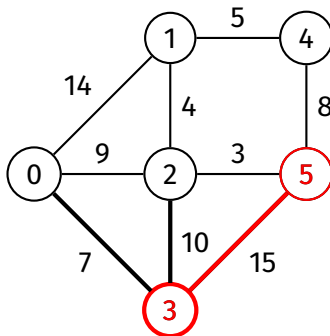
Analysis

Other
Algorithms

Appendix

Example

Relax along (3, 5, 15)
 $\text{dist}[3] + 15 = 22 < \text{dist}[5]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	∞
pred	-1	0	0	0	-1	-1

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

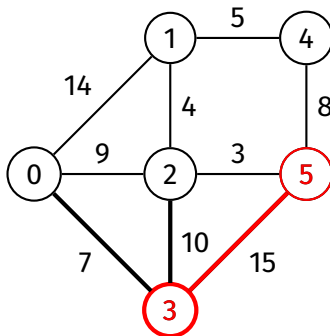
Analysis

Other
Algorithms

Appendix

Example

Relax along (3, 5, 15)
 $\text{dist}[3] + 15 = 22 < \text{dist}[5]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	22
pred	-1	0	0	0	-1	3

Algorithm

Pseudocode

Example

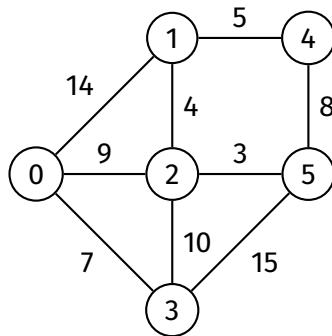
Path Finding

Vertex Set

Analysis

Other
AlgorithmsAppendix
Example

Done with exploring 3



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
        and remove it from vSet
```

```
for each edge ( $v, w, \text{weight}$ ) in  $G$ :  
    relax along ( $v, w, \text{weight}$ )
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	22
pred	-1	0	0	0	-1	3

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

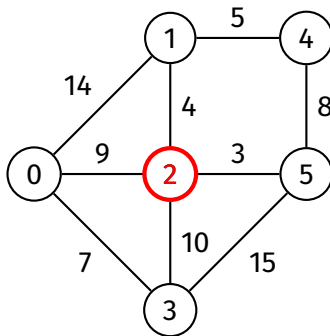
Analysis

Other
Algorithms

Appendix

Example

Remove 2 from vSet



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	22
pred	-1	0	0	0	-1	3

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

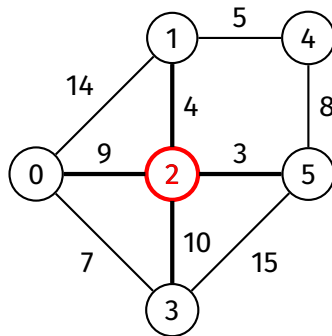
Analysis

Other
Algorithms

Appendix

Example

Explore 2



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
    and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	22
pred	-1	0	0	0	-1	3

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

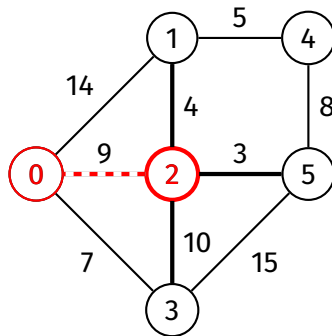
Analysis

Other
Algorithms

Appendix

Example

No need to consider (2, 0, 9)
(0 has already been explored)



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	22
pred	-1	0	0	0	-1	3

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

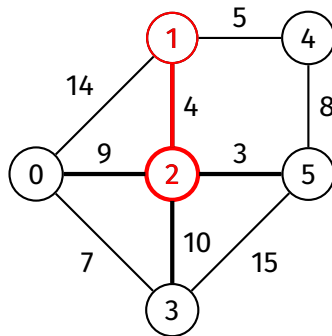
Analysis

Other
Algorithms

Appendix

Example

Relax along (2, 1, 4)



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
        and remove it from vSet
```

```
for each edge ( $v, w, \text{weight}$ ) in  $G$ :  
    relax along ( $v, w, \text{weight}$ )
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	22
pred	-1	0	0	0	-1	3

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

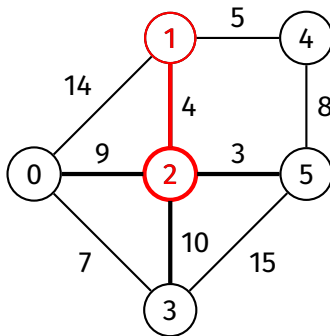
Analysis

Other
Algorithms

Appendix

Example

Relax along (2, 1, 4)
 $\text{dist}[2] + 4$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	22
pred	-1	0	0	0	-1	3

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

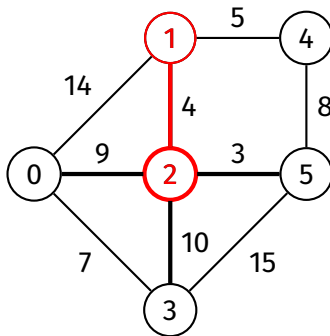
Analysis

Other
Algorithms

Appendix

Example

Relax along (2, 1, 4)
 $\text{dist}[2] + 4 = 13$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	22
pred	-1	0	0	0	-1	3

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

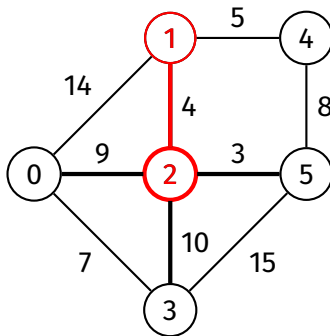
Analysis

Other
Algorithms

Appendix

Example

Relax along (2, 1, 4)
 $\text{dist}[2] + 4 = 13 < \text{dist}[1]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	14	9	7	∞	22
pred	-1	0	0	0	-1	3

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

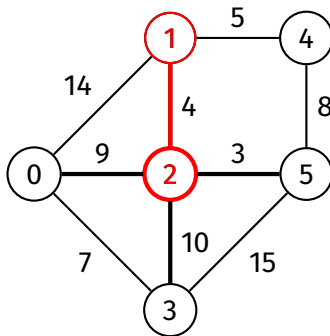
Analysis

Other
Algorithms

Appendix

Example

Relax along (2, 1, 4)
 $\text{dist}[2] + 4 = 13 < \text{dist}[1]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	22
pred	-1	2	0	0	-1	3

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

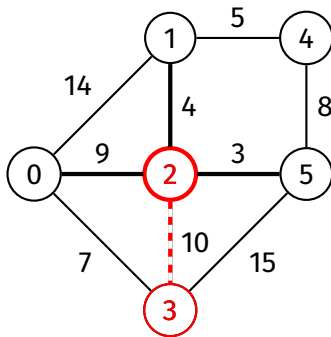
Analysis

Other
Algorithms

Appendix

Example

No need to consider (2, 3, 10)
(3 has already been explored)



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	22
pred	-1	2	0	0	-1	3

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

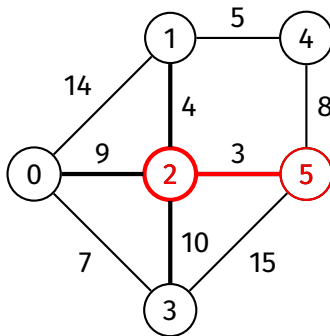
Analysis

Other
Algorithms

Appendix

Example

Relax along (2, 5, 3)



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	22
pred	-1	2	0	0	-1	3

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

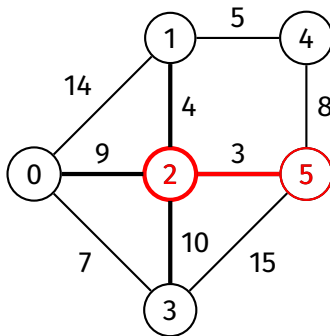
Analysis

Other
Algorithms

Appendix

Example

Relax along (2, 5, 3)
 $\text{dist}[2] + 3$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	22
pred	-1	2	0	0	-1	3

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

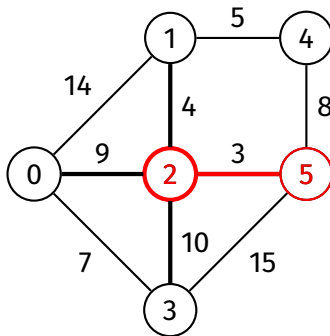
Analysis

Other
Algorithms

Appendix

Example

Relax along (2, 5, 3)
 $\text{dist}[2] + 3 = 12$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	22
pred	-1	2	0	0	-1	3

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

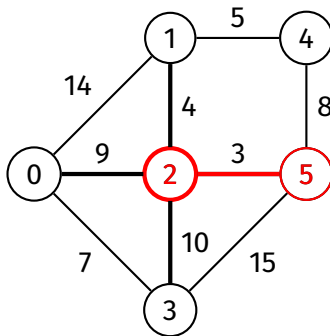
Analysis

Other
Algorithms

Appendix

Example

Relax along (2, 5, 3)
 $\text{dist}[2] + 3 = 12 < \text{dist}[5]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	22
pred	-1	2	0	0	-1	3

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

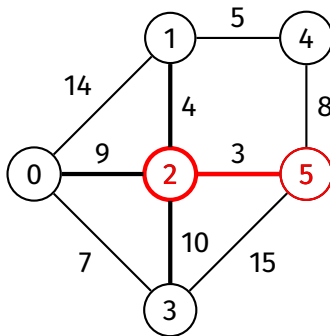
Analysis

Other
Algorithms

Appendix

Example

Relax along (2, 5, 3)
 $\text{dist}[2] + 3 = 12 < \text{dist}[5]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	12
pred	-1	2	0	0	-1	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

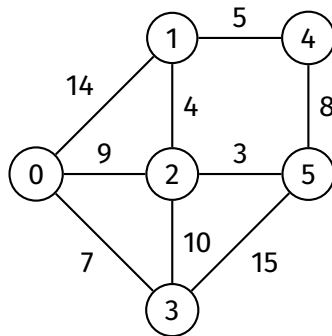
Analysis

Other
Algorithms

Appendix

Example

Done with exploring 2



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	12
pred	-1	2	0	0	-1	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

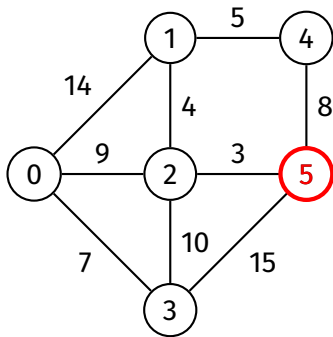
Analysis

Other
Algorithms

Appendix

Example

Remove 5 from vSet



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	12
pred	-1	2	0	0	-1	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

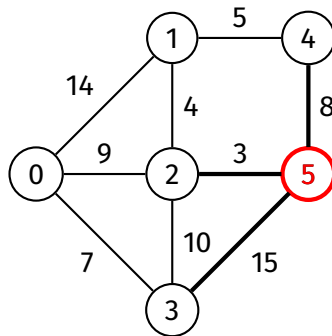
Analysis

Other
Algorithms

Appendix

Example

Explore 5



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	12
pred	-1	2	0	0	-1	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

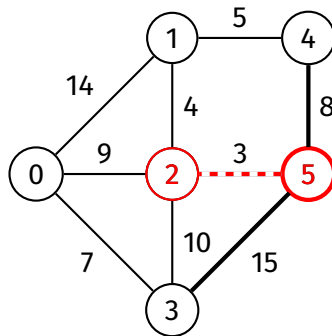
Analysis

Other
Algorithms

Appendix

Example

No need to consider (5, 2, 3)
(2 has already been explored)



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	12
pred	-1	2	0	0	-1	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

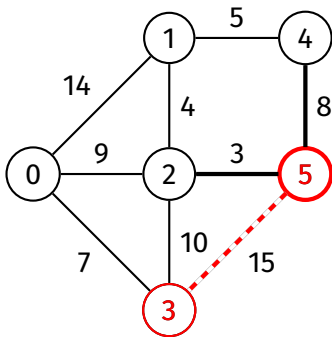
Analysis

Other
Algorithms

Appendix

Example

No need to consider (5, 3, 15)
(3 has already been explored)



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	12
pred	-1	2	0	0	-1	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

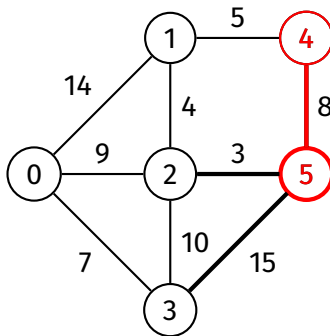
Analysis

Other
Algorithms

Appendix

Example

Relax along (5, 4, 8)



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
        and remove it from vSet
```

```
for each edge ( $v, w, \text{weight}$ ) in  $G$ :  
    relax along ( $v, w, \text{weight}$ )
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	12
pred	-1	2	0	0	-1	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

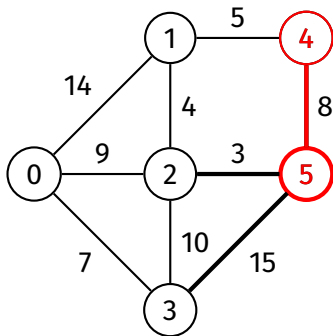
Analysis

Other
Algorithms

Appendix

Example

Relax along (5, 4, 8)
 $\text{dist}[5] + 8$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	12
pred	-1	2	0	0	-1	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

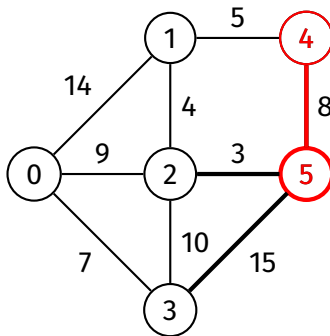
Analysis

Other
Algorithms

Appendix

Example

Relax along (5, 4, 8)
 $\text{dist}[5] + 8 = 20$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	12
pred	-1	2	0	0	-1	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

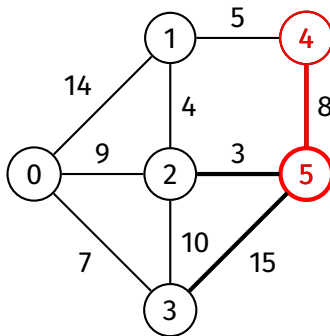
Analysis

Other
Algorithms

Appendix

Example

Relax along (5, 4, 8)
 $\text{dist}[5] + 8 = 20 < \text{dist}[4]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	∞	12
pred	-1	2	0	0	-1	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

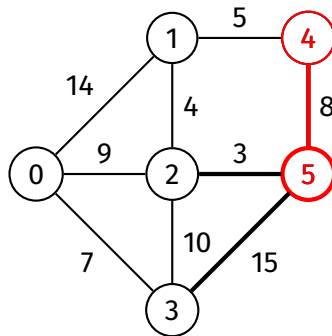
Analysis

Other
Algorithms

Appendix

Example

Relax along (5, 4, 8)
 $\text{dist}[5] + 8 = 20 < \text{dist}[4]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	20	12
pred	-1	2	0	0	5	2

Algorithm

Pseudocode

Example

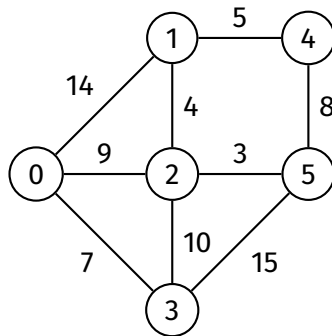
Path Finding

Vertex Set

Analysis

Other
AlgorithmsAppendix
Example

Done with exploring 5



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	20	12
pred	-1	2	0	0	5	2

Algorithm

Pseudocode

Example

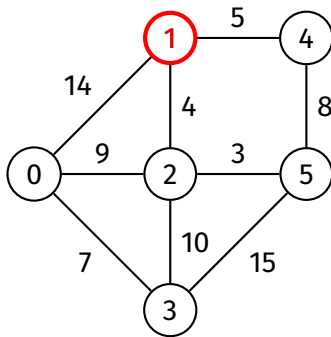
Path Finding

Vertex Set

Analysis

Other
AlgorithmsAppendix
Example

Remove 1 from vSet



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	20	12
pred	-1	2	0	0	5	2

Algorithm

Pseudocode

Example

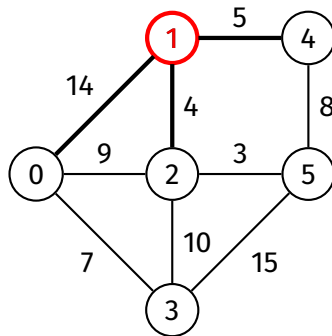
Path Finding

Vertex Set

Analysis

Other
AlgorithmsAppendix
Example

Explore 1



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	20	12
pred	-1	2	0	0	5	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

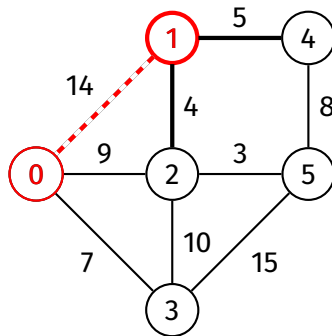
Analysis

Other
Algorithms

Appendix

Example

No need to consider (1, 0, 14)
(0 has already been explored)



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	20	12
pred	-1	2	0	0	5	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

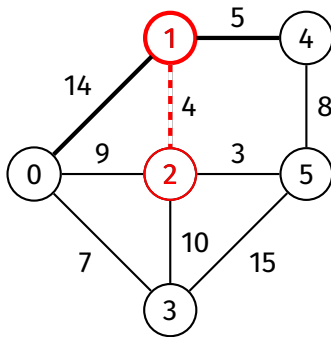
Analysis

Other
Algorithms

Appendix

Example

No need to consider (1, 2, 4)
(2 has already been explored)



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	20	12
pred	-1	2	0	0	5	2

Algorithm

Pseudocode

Example

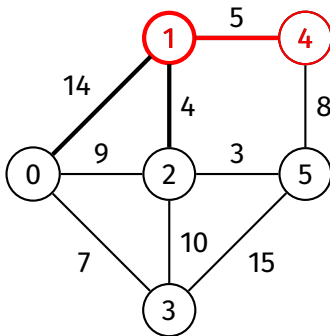
Path Finding

Vertex Set

Analysis

Other
AlgorithmsAppendix
Example

Relax along (1, 4, 5)



```
while vSet is not empty:
    find vertex  $v$  in vSet such that
         $\text{dist}[v]$  is minimal
    and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	20	12
pred	-1	2	0	0	5	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

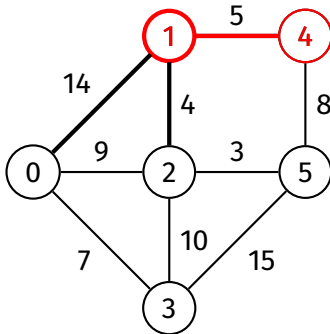
Analysis

Other
Algorithms

Appendix

Example

Relax along (1, 4, 5)
 $\text{dist}[1] + 5$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	20	12
pred	-1	2	0	0	5	2

Algorithm

Pseudocode

Example

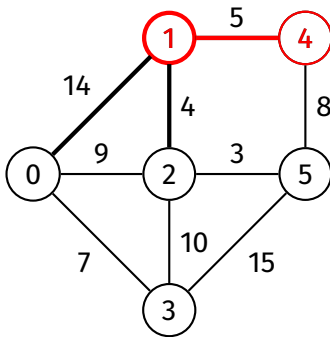
Path Finding

Vertex Set

Analysis

Other
AlgorithmsAppendix
Example

Relax along (1, 4, 5)
 $\text{dist}[1] + 5 = 18$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	20	12
pred	-1	2	0	0	5	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

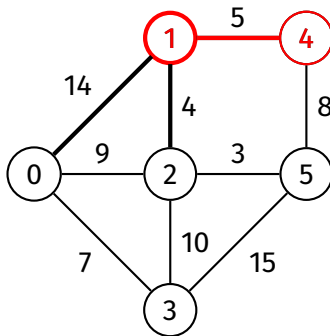
Analysis

Other
Algorithms

Appendix

Example

Relax along (1, 4, 5)
 $\text{dist}[1] + 5 = 18 < \text{dist}[4]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	20	12
pred	-1	2	0	0	5	2

Algorithm

Pseudocode

Example

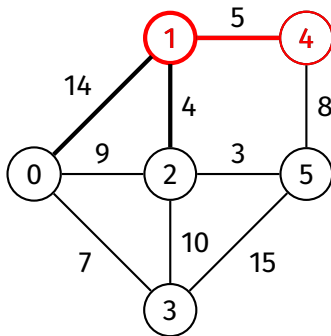
Path Finding

Vertex Set

Analysis

Other
AlgorithmsAppendix
Example

Relax along (1, 4, 5)
 $\text{dist}[1] + 5 = 18 < \text{dist}[4]$



while vSet is not empty:
 find vertex v in vSet such that
 $\text{dist}[v]$ is minimal
 and remove it from vSet

for each edge (v, w, weight) in G :
 relax along (v, w, weight)

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	18	12
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

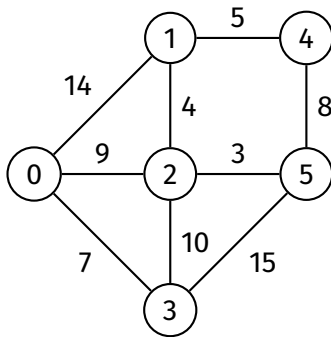
Analysis

Other
Algorithms

Appendix

Example

Done with exploring 1



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	18	12
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

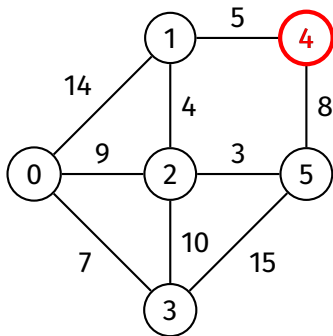
Analysis

Other
Algorithms

Appendix

Example

Remove 4 from vSet



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	18	12
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

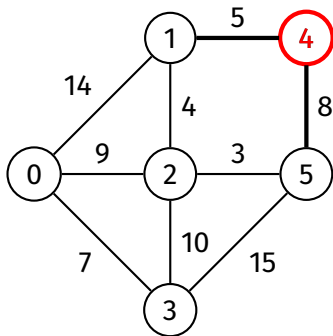
Path Finding

Vertex Set

Analysis

Other
AlgorithmsAppendix
Example

Explore 4



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	18	12
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

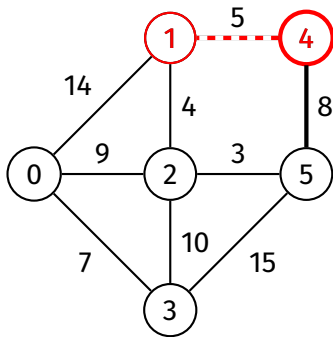
Analysis

Other
Algorithms

Appendix

Example

No need to consider (4, 1, 5)
(1 has already been explored)



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	18	12
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

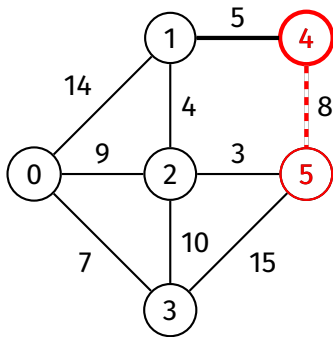
Analysis

Other
Algorithms

Appendix

Example

No need to consider (4, 5, 8)
(5 has already been explored)



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	18	12
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

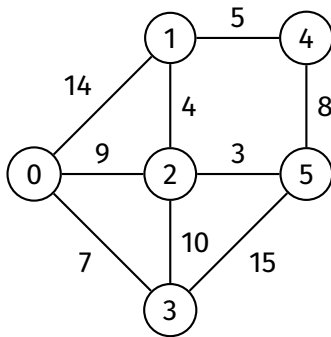
Path Finding

Vertex Set

Analysis

Other
AlgorithmsAppendix
Example

Done with exploring 4



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
         $\text{dist}[v]$  is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	18	12
pred	-1	2	0	0	1	2

Algorithm

Pseudocode

Example

Path Finding

Vertex Set

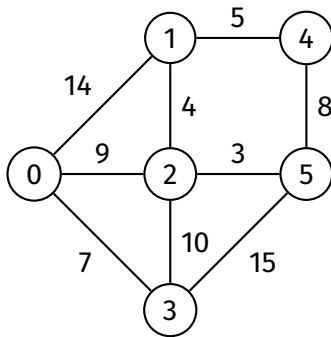
Analysis

Other
Algorithms

Appendix

Example

Finished



```
while vSet is not empty:  
    find vertex  $v$  in vSet such that  
        dist[ $v$ ] is minimal  
        and remove it from vSet
```

```
for each edge  $(v, w, \text{weight})$  in  $G$ :  
    relax along  $(v, w, \text{weight})$ 
```

	[0]	[1]	[2]	[3]	[4]	[5]
dist	0	13	9	7	18	12
pred	-1	2	0	0	1	2