Example and proof of Dijkstra's Algorithm

Hao WU

CSE
The Chinese University of Hong Kong

Adapted from the slides of the previous offerings of the course.

Dijkstra's Algorithm

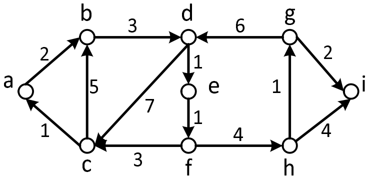
The algorithm solves the single-source shortest-paths (SSSP) problem on a directed graph $G = (V, E)$ with positive edge weights.

Let $V' \subseteq V$ be the current set of vertices whose shortest paths from the source vertex s have been found and $S = V \backslash V'$.

The crucial part of the algorithm is the edge relaxation idea. Essentially, we will prove this later, it is to maintain, for each $v \in S$, the "current shortest" distance from s only through the vertices in V'.

Example

Suppose that the source vertex is a.

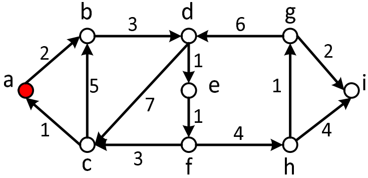


vertex v	$dist(v)$	$parent(v)$
a	0	nil
b	∞	nil
c	∞	nil
d	∞	nil
e	∞	nil
f	∞	nil
g	∞	nil
h	∞	nil
i	∞	nil

$V' = \emptyset$ and
$S = \{a, b, c, d, e, f, g, h, i\}$.
Since $dist(a)$ is the smallest among those of vertices in S, pick a.

Example

Relax the out-going edges of a:

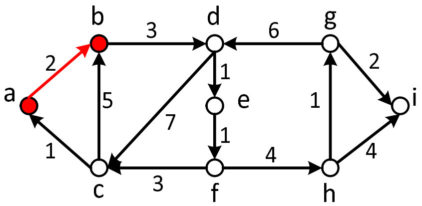


vertex v	$dist(v)$	$parent(v)$
a	0	nil
b	$\infty \to 2$	nil $\to$ a
c	∞	nil
d	∞	nil
e	∞	nil
f	∞	nil
g	∞	nil
h	∞	nil
i	∞	nil

$V' = \{a\}$ and
$S = \{b, c, d, e, f, g, h, i\}$.
The "current shortest" distance of b from a only through the vertices in V' is updated. After then, $dist(b)$ is the smallest among those of vertices in S. Pick b.

Example

Relax the out-going edges of b:

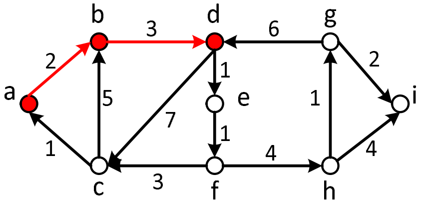


vertex v	$dist(v)$	$parent(v)$
a	0	nil
b	2	a
c	∞	nil
d	$\infty \rightarrow 5$	nil $\rightarrow$ b
e	∞	nil
f	∞	nil
g	∞	nil
h	∞	nil
i	∞	nil

$V' = \{a, b\}$ and
$S = \{c, d, e, f, g, h, i\}$.
Similarly, update the "current shortest" distance of d from a only through the vertices in V'. And $dist(d)$ is the smallest among those in S. Pick d.

Example

Relax the out-going edges of d:

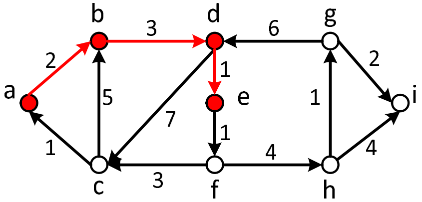


vertex v	$dist(v)$	$parent(v)$
a	0	nil
b	2	a
c	$\infty \to 12$	nil $\to d$
d	5	b
e	$\infty \to 6$	nil $\to d$
f	∞	nil
g	∞	nil
h	∞	nil
i	∞	nil

$V' = \{a, b, d\}$ and
$S = \{c, e, f, g, h, i\}$.
Since after the updates, $dist(e)$ is the smallest among those in S, pick e.

Example

Relax the out-going edges of e:

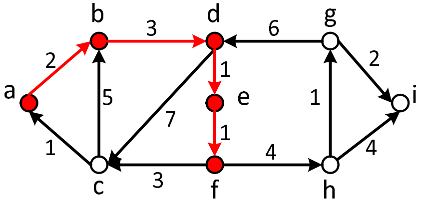


vertex v	$dist(v)$	$parent(v)$
a	0	nil
b	2	a
c	12	d
d	5	b
e	6	d
f	$\infty \to 7$	nil $\to e$
g	∞	nil
h	∞	nil
i	∞	nil

$V' = \{a, b, d, e\}$ and
$S = \{c, f, g, h, i\}$.

Example

Relax the out-going edges of f:

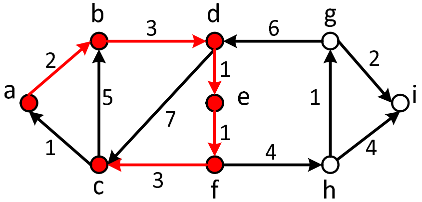


vertex v	$dist(v)$	$parent(v)$
a	0	nil
b	2	a
c	$12 \rightarrow 10$	$d \rightarrow f$
d	5	b
e	6	d
f	7	e
g	∞	nil
h	$\infty \rightarrow 11$	nil $\rightarrow f$
i	∞	nil

$V' = \{a, b, d, e, f\}$ and
$S = \{c, g, h, i\}$.

Example

Relax the out-going edges of c:

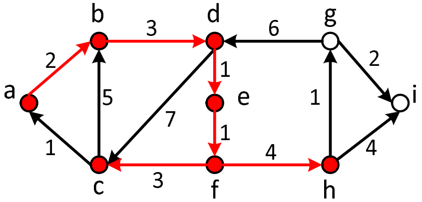


vertex v	$dist(v)$	$parent(v)$
a	0	nil
b	2	a
c	10	f
d	5	b
e	6	d
f	7	e
g	∞	nil
h	11	f
i	∞	nil

$V' = \{a, b, c, d, e, f\}$ and
$S = \{g, h, i\}$.

Example

Relax the out-going edges of h:

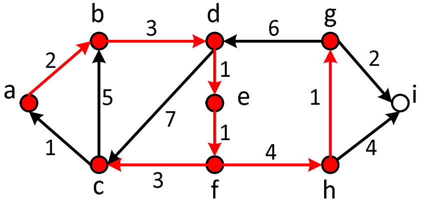


vertex v	$dist(v)$	$parent(v)$
a	0	nil
b	2	a
c	10	f
d	5	b
e	6	d
f	7	e
g	$\infty \to 12$	nil $\to h$
h	11	f
i	$\infty \to 15$	nil $\to h$

$V' = \{a, b, c, d, e, f, h\}$ and
$S = \{g, i\}$.

Example

Relax the out-going edges of g:

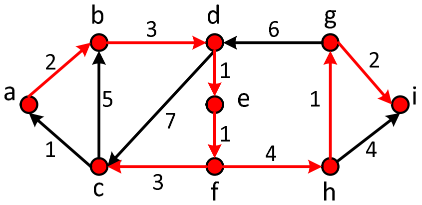


vertex v	$dist(v)$	$parent(v)$
a	0	nil
b	2	a
c	10	f
d	5	b
e	6	d
f	7	e
g	12	h
h	11	f
i	$15 \to 14$	$h \to g$

$V' = \{a, b, c, d, e, f, g, h\}$ and $S = \{i\}$.

Example

Relax the out-going edges of i:



vertex v	$dist(v)$	$parent(v)$
a	0	nil
b	2	a
c	10	f
d	5	b
e	6	d
f	7	e
g	12	h
h	11	f
i	14	g

$V' = \{a, b, c, d, e, f, g, h, i\}$ and
$S = \{\}$.
Done.

Correctness of Dijkstra's Algorithm

Lemma: When vertex v is removed from S, $dist(v)$ equals precisely the shortest path distance—denoted as $spdist(v)$—from s to v.

The correctness of Dijkstra's algorithm follows from the lemma.

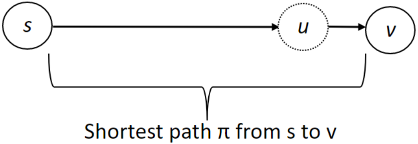
Correctness of Dijkstra's Algorithm

We will prove the claim by induction on the sequence of vertices removed.

- Base case:
  This is obviously true for the first vertex removed, which is s itself with $dist(s) = 0$.
- Inductive:
  Assume the claim is true with respect to all the vertices already removed. Let v be the next node to be removed. Need to prove $dist(v) = spdist(v)$.

Correctness of Dijkstra's Algorithm

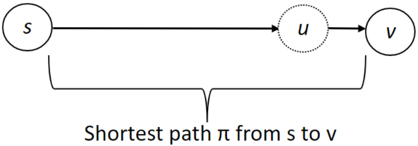
Consider an arbitrary shortest path π from s to v. Let u be the vertex right before v on π.



Claim: u must have been removed from S.

Our target lemma follows from the above claim because, by our inductive assumption, $dist(u) = spdist(u)$ when u was removed. Then, the algorithm relaxed the edge (u, v), which must have set $dist(v) = spdist(u) + w(u, v) = spdist(v)$.

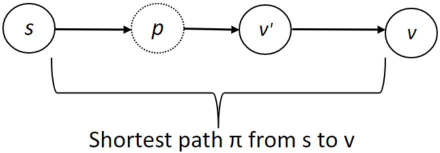
Correctness of Dijkstra's Algorithm



Stronger claim: All the nodes on π from s to u must have been removed.

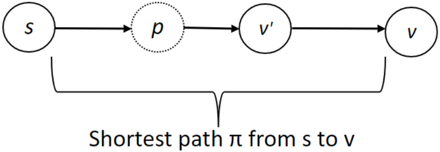
Correctness of Dijkstra's Algorithm

We will prove the stronger claim by contradiction.



Suppose the statement is not true. When v is to be removed from S, another vertex on π — let it be v' — still remains in S. Define p as the vertex right before v' on π.

Correctness of Dijkstra's Algorithm



By the inductive assumption, $dist(p) = spdist(p)$ when p was removed. Hence, after relaxing the edge (p, v'), we have $dist(v') = spdist(p) + w(p, v') = spdist(v')$.

But this means $dist(v') = spdist(v') < spdist(v) \leq dist(v)$!

Hence, v' should be the next vertex to be removed from S, contradicting the definition of v.