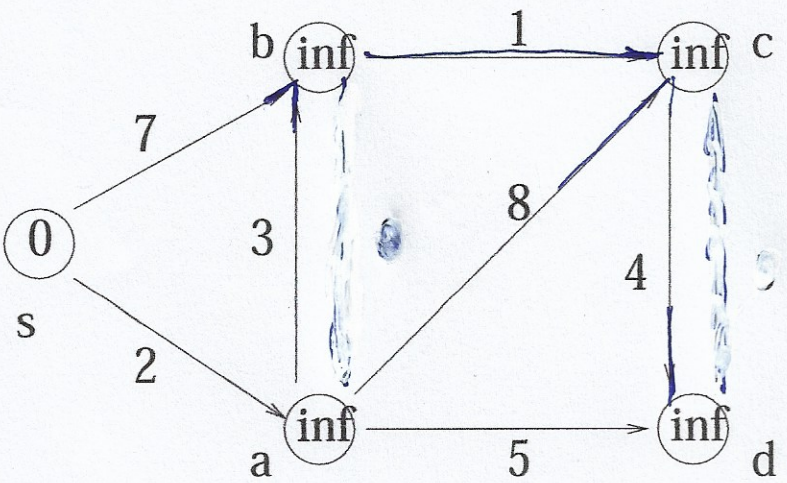
Prim ~~Dijkstra~~'s Algorithm

Example:



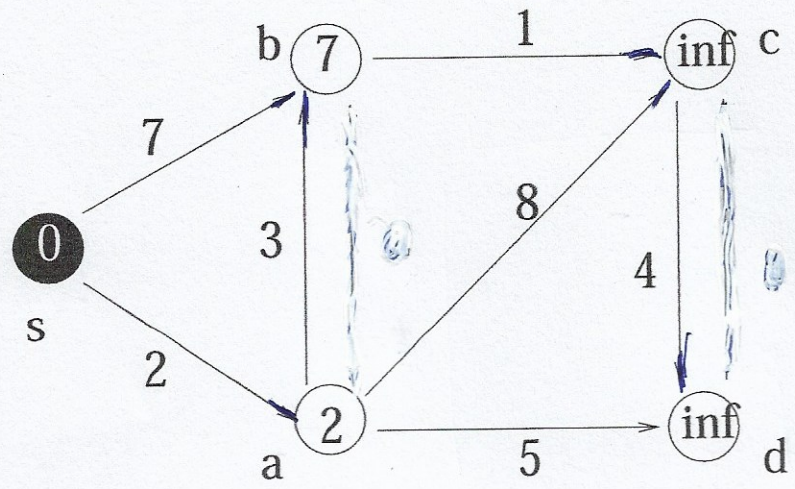
Step 0: Initialization.

v	s	a	b	c	d
$d[v]$	0	∞	∞	∞	∞
$pred[v]$	nil	nil	nil	nil	nil
$color[v]$	W	W	W	W	W

Priority Queue:

v	s	a	b	c	d
$d[v]$	0	∞	∞	∞	∞

Example:



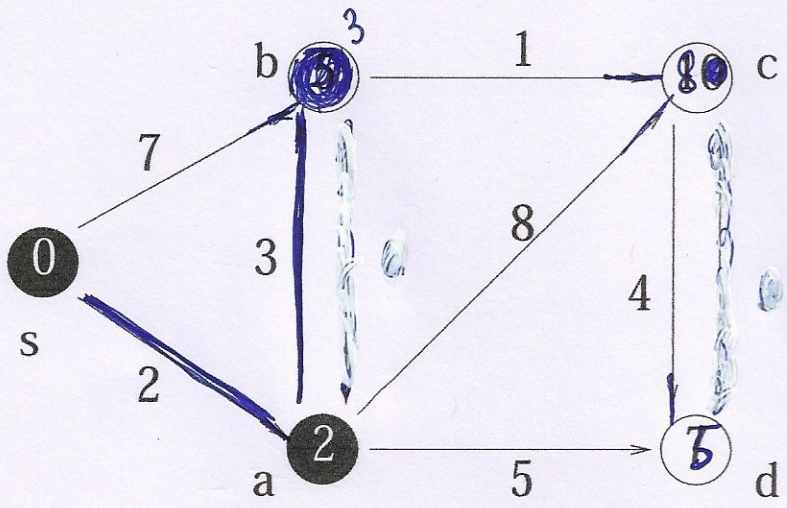
Step 1: As $Adj[s] = \{a, b\}$, work on a and b and update information.

v	s	a	b	c	d
$d[v]$	0	2	7	∞	∞
$pred[v]$	nil	s	s	nil	nil
$color[v]$	B	W	W	W	W

Priority Queue:

v	a	b	c	d
$d[v]$	2	7	∞	∞

Example:



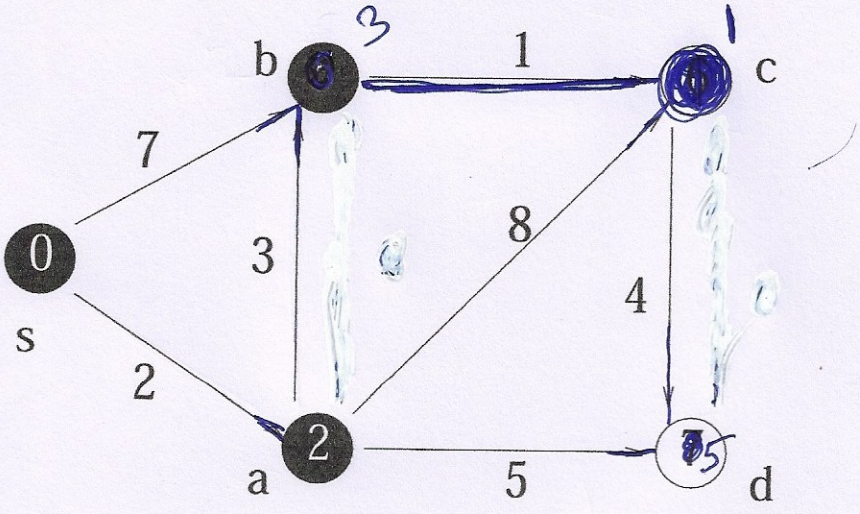
Step 2: After Step 1, a has the minimum key in the priority queue. As $Adj[a] = \{b, c, d\}$, work on b, c, d and update information.

v	s	a	b	c	d
$d[v]$	0	2	3	8	5
$pred[v]$	nil	s	a	a	a
$color[v]$	B	B	W	W	W

Priority Queue:

v	b	c	d
$d[v]$	3	8	5

Example:



Step 3: After Step 2, b has the minimum key in the priority queue. As $Adj[b]$ $\{a, c\}$, work on c and update information.

v	s	a	b	c	d
$d[v]$	0	2	3	1	5
$pred[v]$	nil	s	a	b	a
$color[v]$	B	B	B	W	W

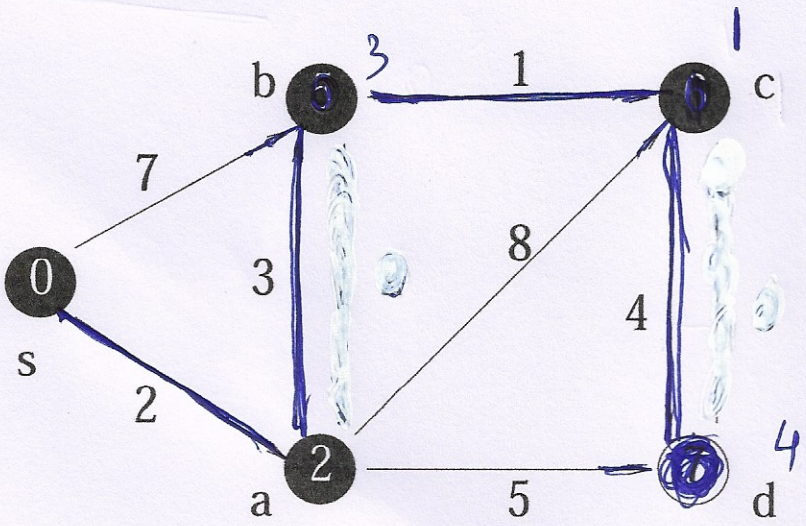
Priority Queue:

v	c	d
$d[v]$	1	5

Prim's

~~Dijkstra's~~ Algorithm

Example:



Step 4: After Step 3, c has the minimum key in the priority queue. As $Adj[c] = \{d\}$, work on d and update information.

v	s	a	b	c	d
$d[v]$	0	2	3	1	4
$pred[v]$	nil	s	a	b	a c
$color[v]$	B	B	B	B	W

better edge found so pred is changed.

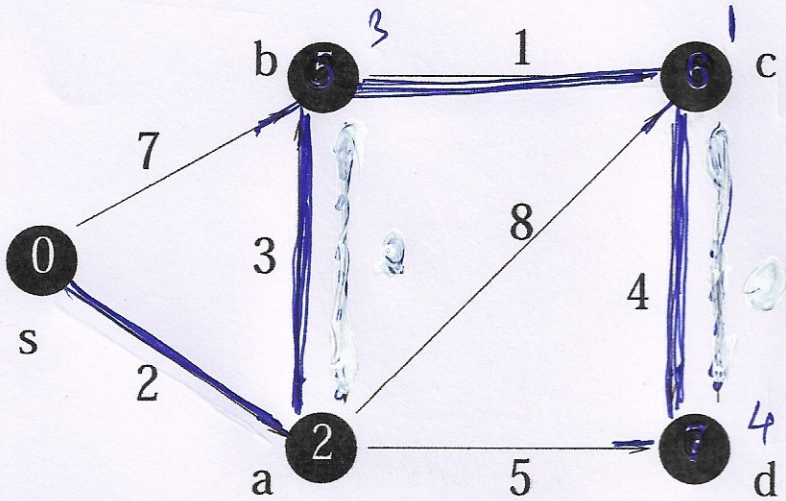
$w(c,d) < key[d]$

Priority Queue:

v	d
$d[v]$	4

Dijkstra's Algorithm

Example:



Step 5: After Step 4, d has the minimum key in the priority queue. As $Adj[d] = \{\bullet\}$, [illegible].

v	s	a	b	c	d
$d[v]$	0	2	3	1	4
$pred[v]$	nil	s	a	b	c
$color[v]$	B	B	B	B	B

Priority Queue: $Q = \emptyset$.

We are done.