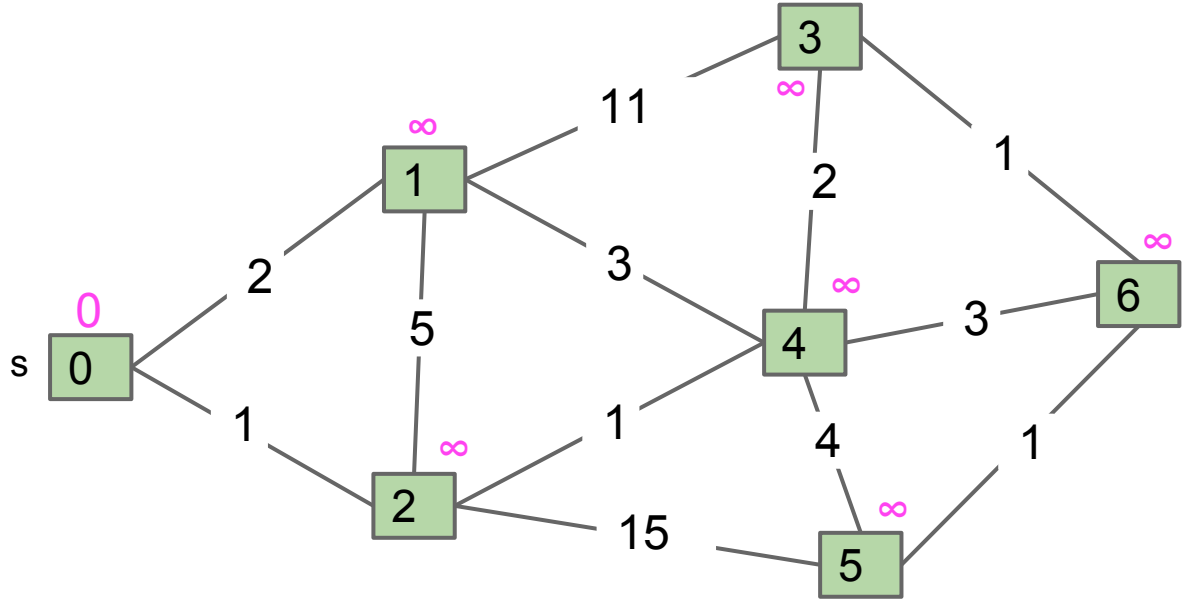


Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0	0	-
1	∞	-
2	∞	-
3	∞	-
4	∞	-
5	∞	-
6	∞	-

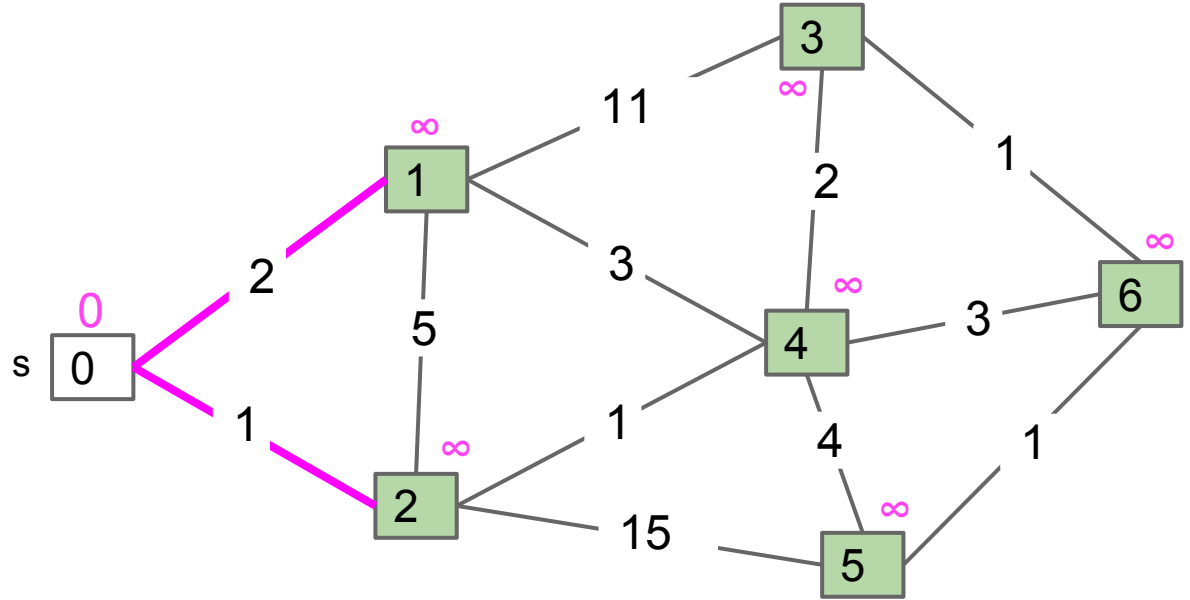


Fringe: $[(0: 0), (1: \infty), (2: \infty), (3: \infty), (4: \infty), (5: \infty), (6: \infty)]$

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0	0	-
1	∞	-
2	∞	-
3	∞	-
4	∞	-
5	∞	-
6	∞	-

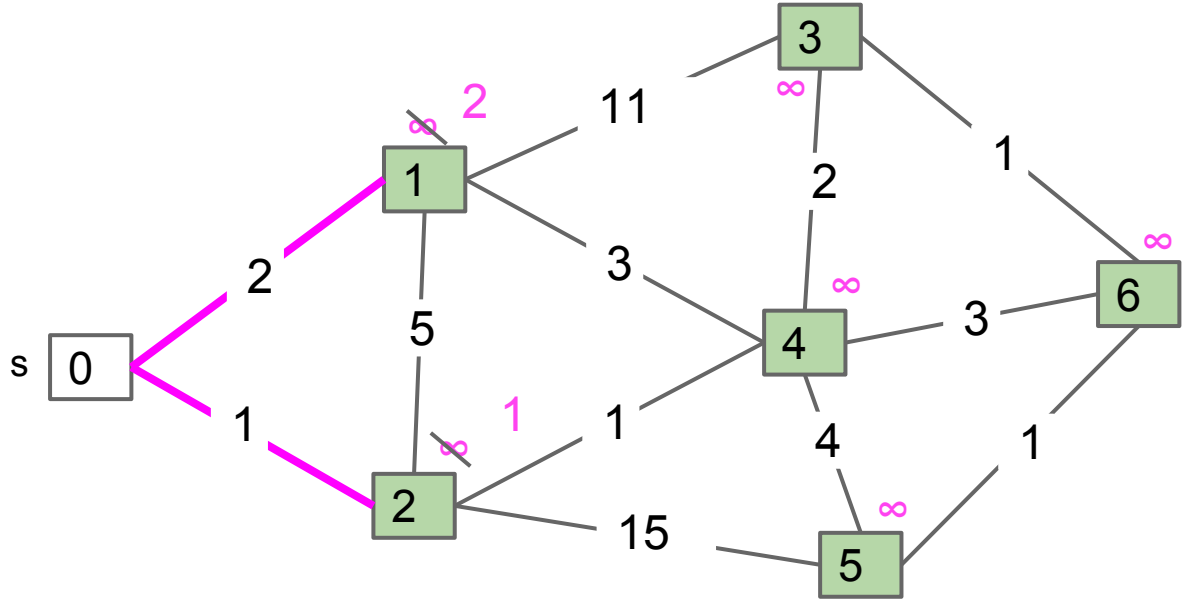


Fringe: [(1: ∞), (2: ∞), (3: ∞), (4: ∞), (5: ∞), (6: ∞)]

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0		-
1	2	0
2	1	0
3	∞	-
4	∞	-
5	∞	-
6	∞	-

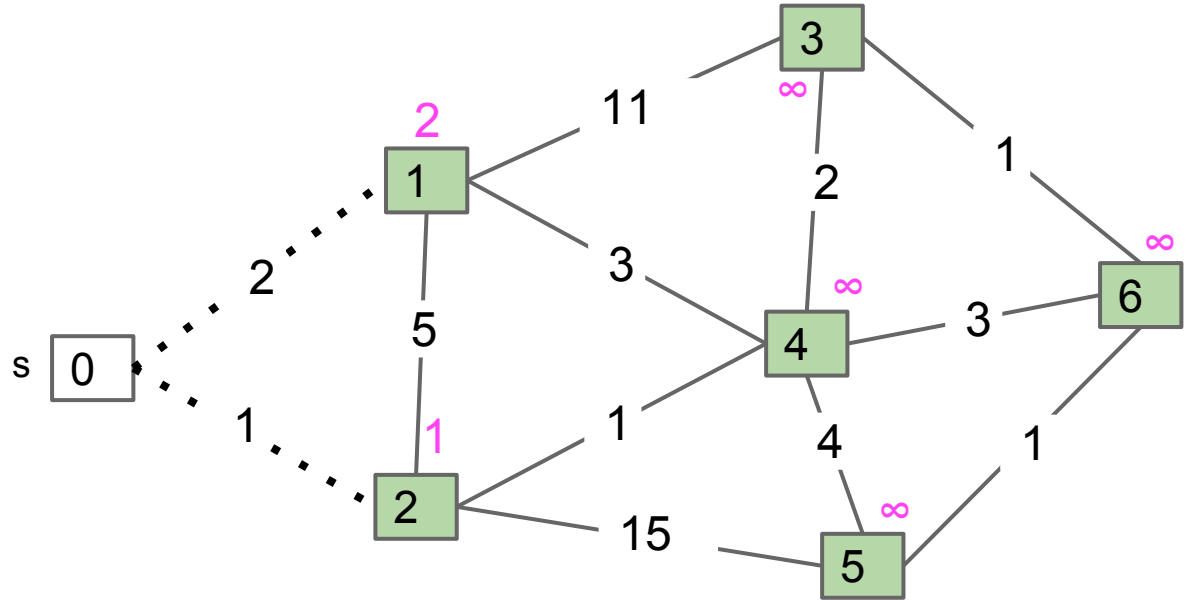


Fringe: [(2: 1), (1: 2), (3: ∞), (4: ∞), (5: ∞), (6: ∞)]

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0	0	-
1	2	0
2	1	0
3	∞	-
4	∞	-
5	∞	-
6	∞	-

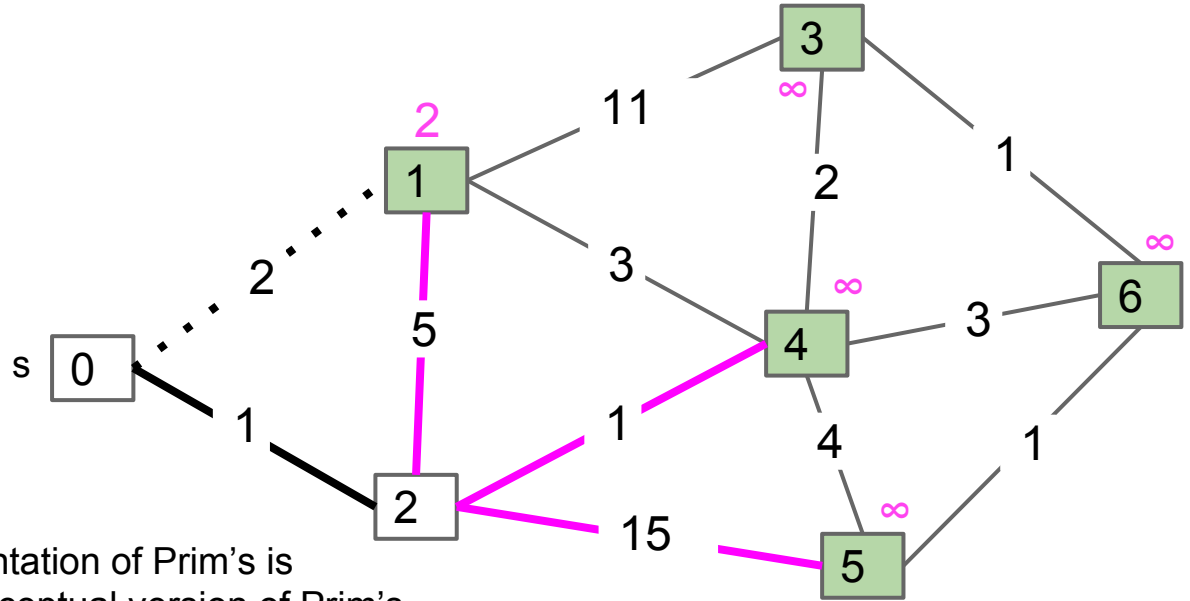


Fringe: [(2: 1), (1: 2), (3: ∞), (4: ∞), (5: ∞), (6: ∞)]

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0		-
1	2	0
2		0
3	∞	-
4	∞	-
5	∞	-
6	∞	-



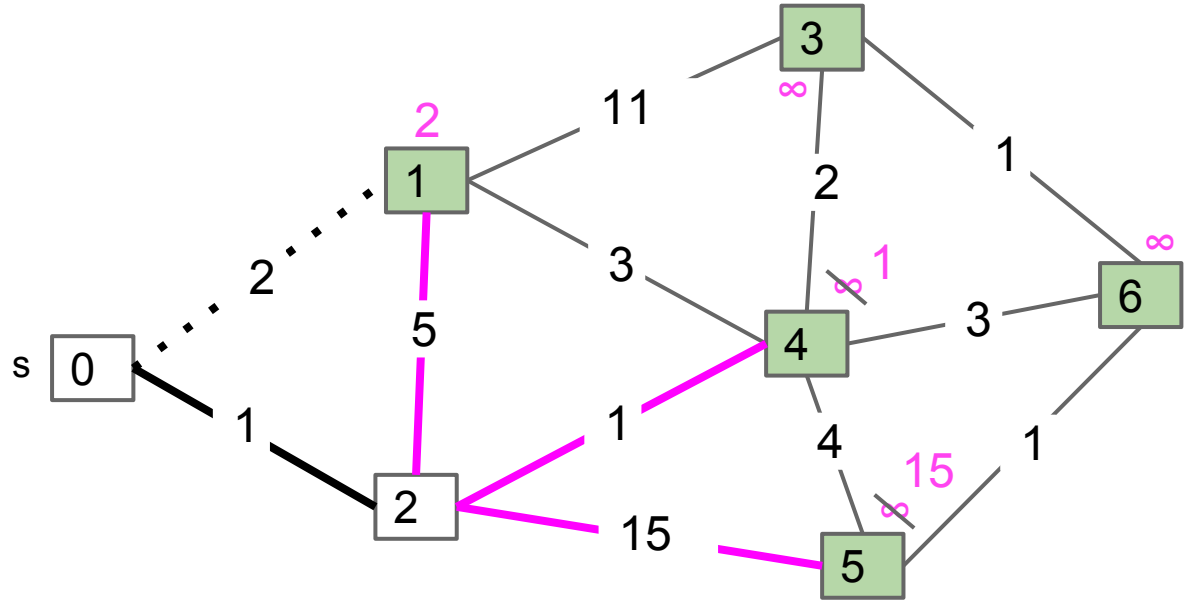
Note: Vertex removal in this implementation of Prim's is equivalent to edge addition in the conceptual version of Prim's.

Fringe: [(1: 2), (3: ∞), (4: ∞), (5: ∞), (6: ∞)]

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0		-
1	2	0
2		0
3	∞	-
4	1	2
5	15	2
6	∞	-

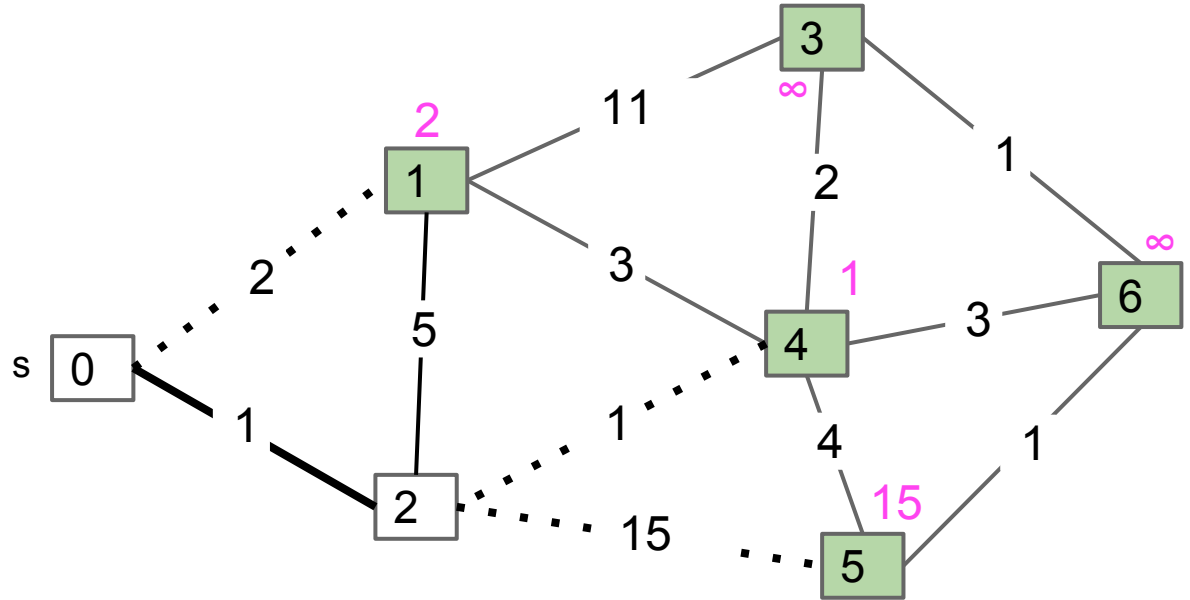


Fringe: [(4: 1), (1: 2), (5: 15), (3: ∞), (6: ∞)]

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0		-
1	2	0
2		0
3	∞	-
4	1	2
5	15	2
6	∞	-



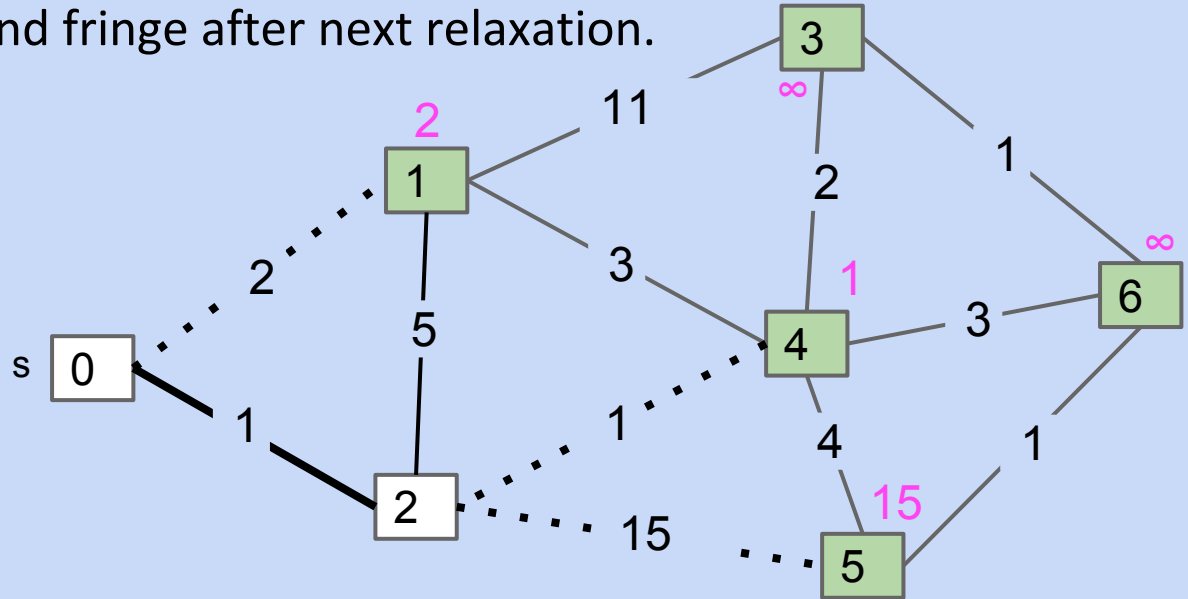
Fringe: [(4: 1), (1: 2), (5: 15), (3: ∞), (6: ∞)]

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

- Show distTo , edgeTo , and fringe after next relaxation.

#	distTo	edgeTo
0		-
1	2	0
2		0
3	∞	-
4	1	2
5	15	2
6	∞	-

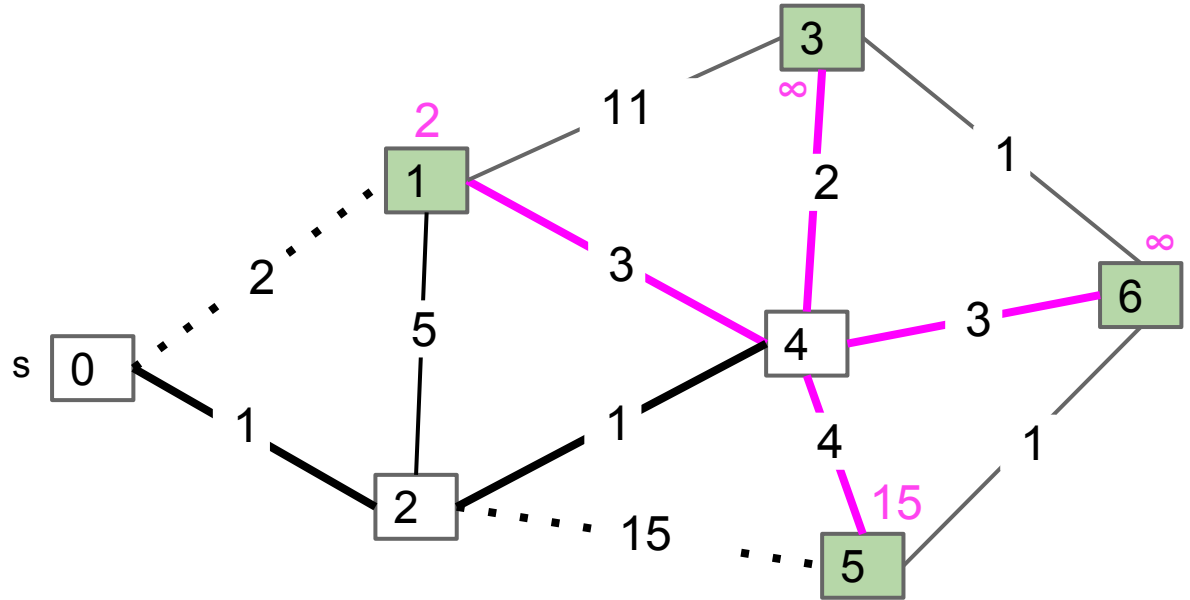


Fringe: [(4: 1), (1: 2), (5: 15), (3: ∞), (6: ∞)]

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0		-
1	2	0
2		0
3	∞	-
4		2
5	15	2
6	∞	-

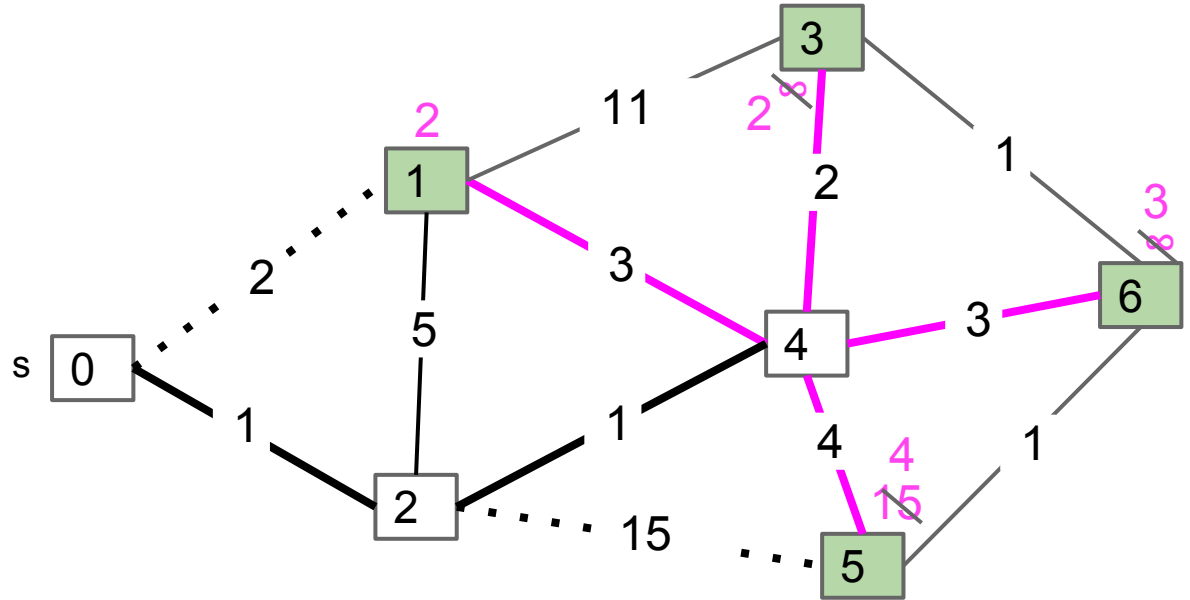


Fringe: [(1: 2), (5: 15), (3: ∞), (6: ∞)]

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0	-	-
1	2	0
2	0	0
3	2	4
4	2	2
5	4	4
6	3	4

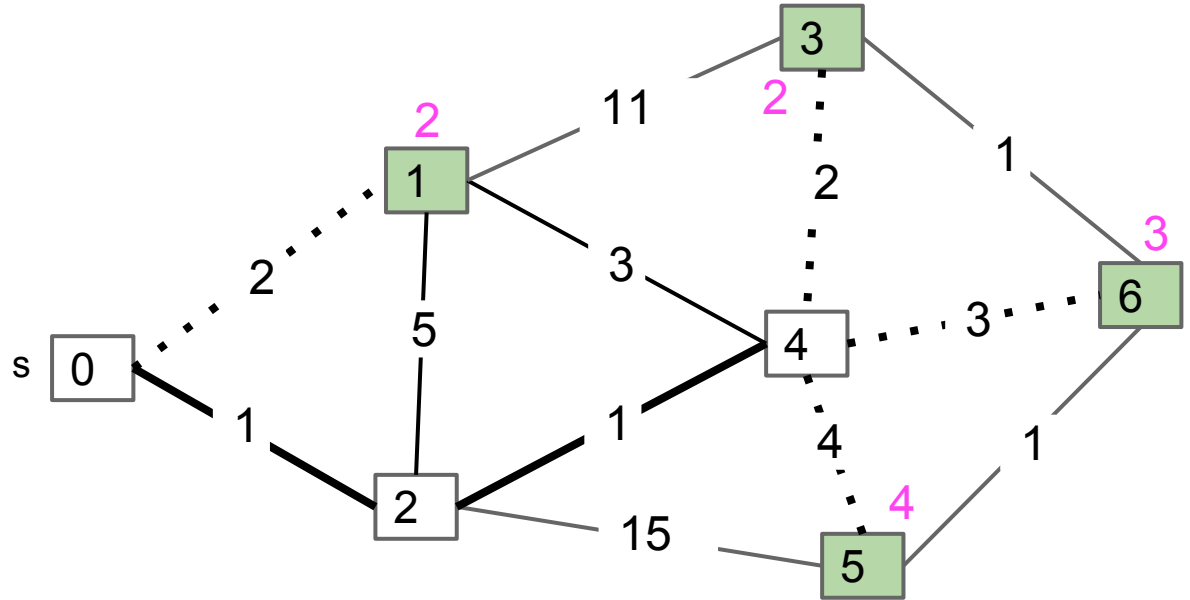


Fringe: [(1: 2), (3: 2), (6: 3), (5: 4)]

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0		-
1	2	0
2		0
3	2	4
4		2
5	4	4
6	3	4

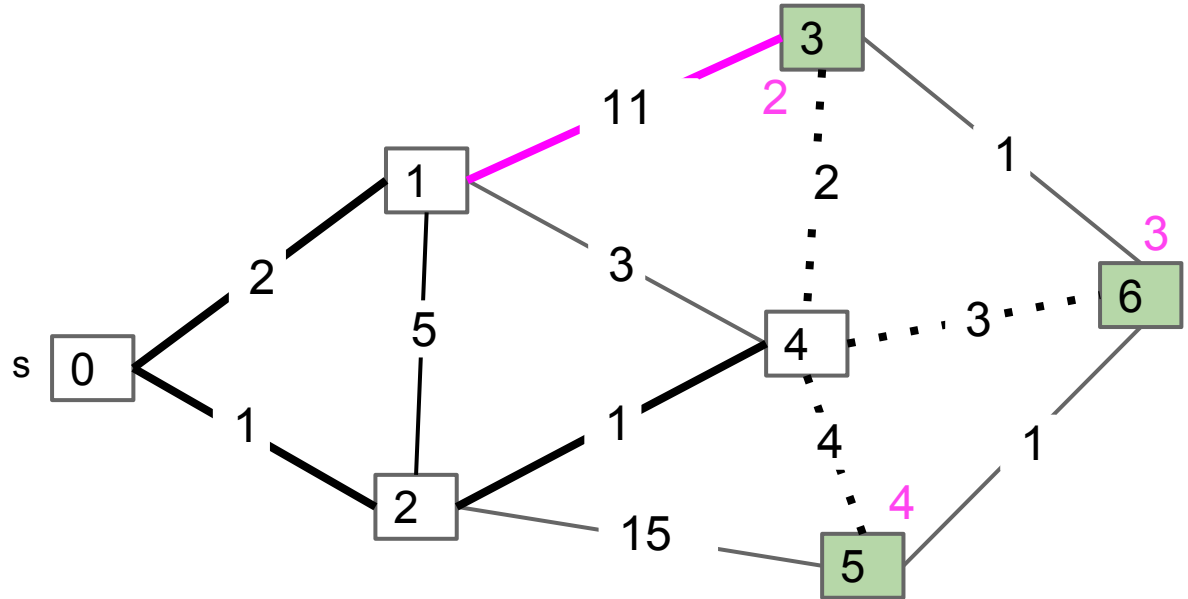


Fringe: [(1: 2), (3: 2), (6: 3), (5: 4)]

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0	-	-
1	0	0
2	0	0
3	2	4
4	2	2
5	4	4
6	3	4



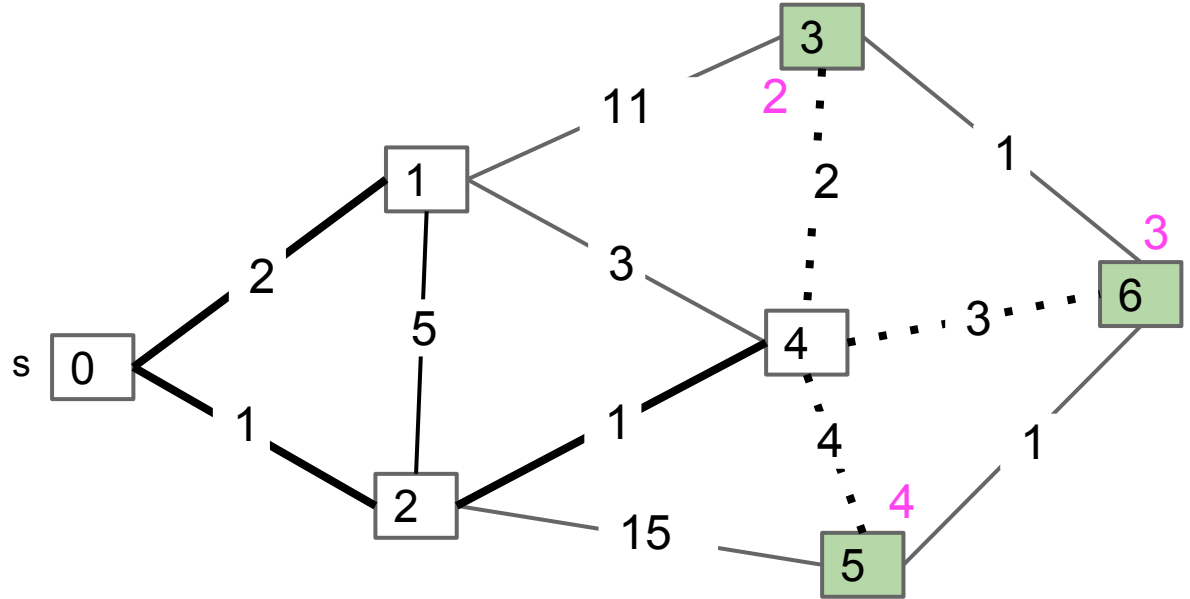
Fringe: [(3: 2), (6: 3), (5: 4)]

No need to consider edges with weight 5 and 3 since other side is already marked!

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0	-	-
1	0	0
2	0	0
3	2	4
4	2	2
5	4	4
6	3	4

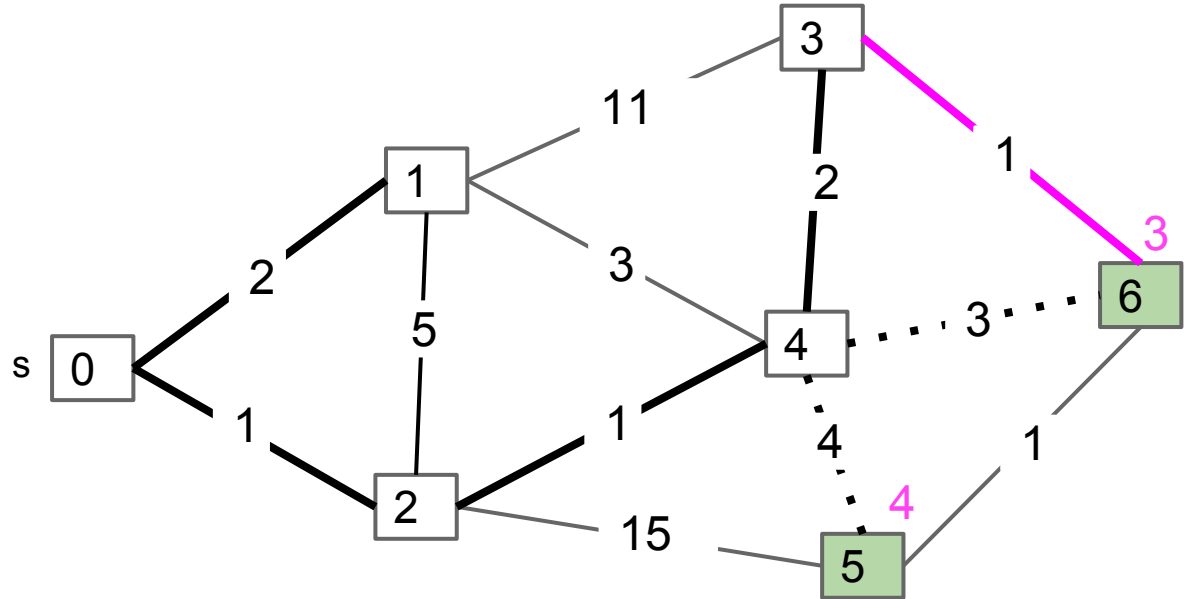


Fringe: [(3: 2), (6: 3), (5: 4)]

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0		-
1		0
2		0
3		4
4		2
5	4	4
6	3	4

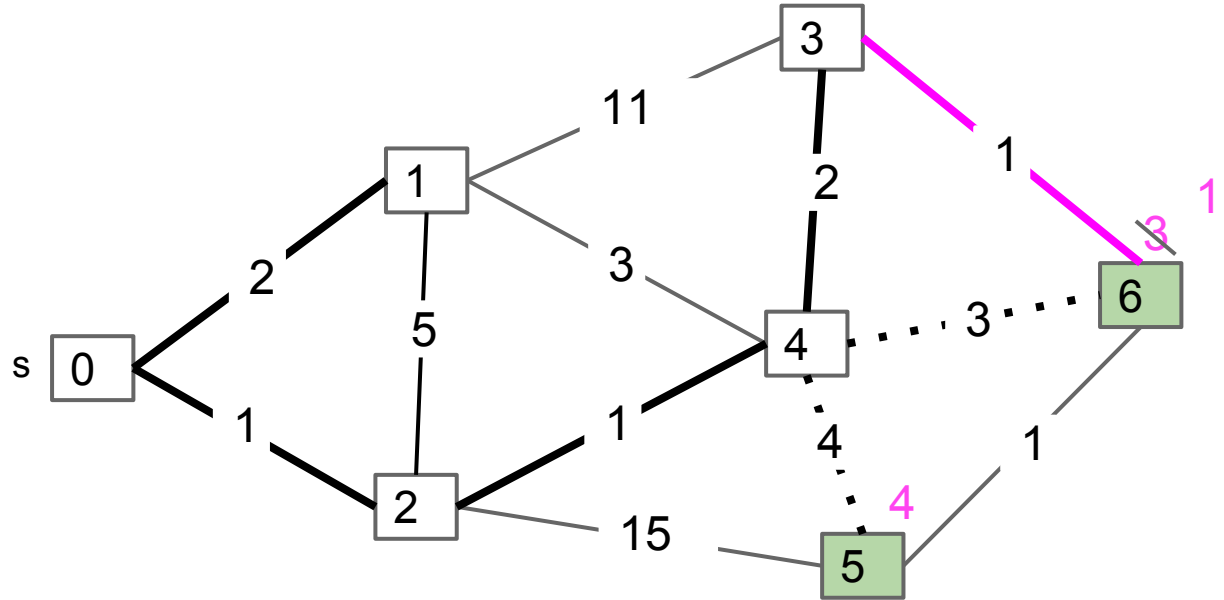


Fringe: [(6: 3), (5: 4)]

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0		-
1		0
2		0
3		4
4		2
5	4	4
6	1	3

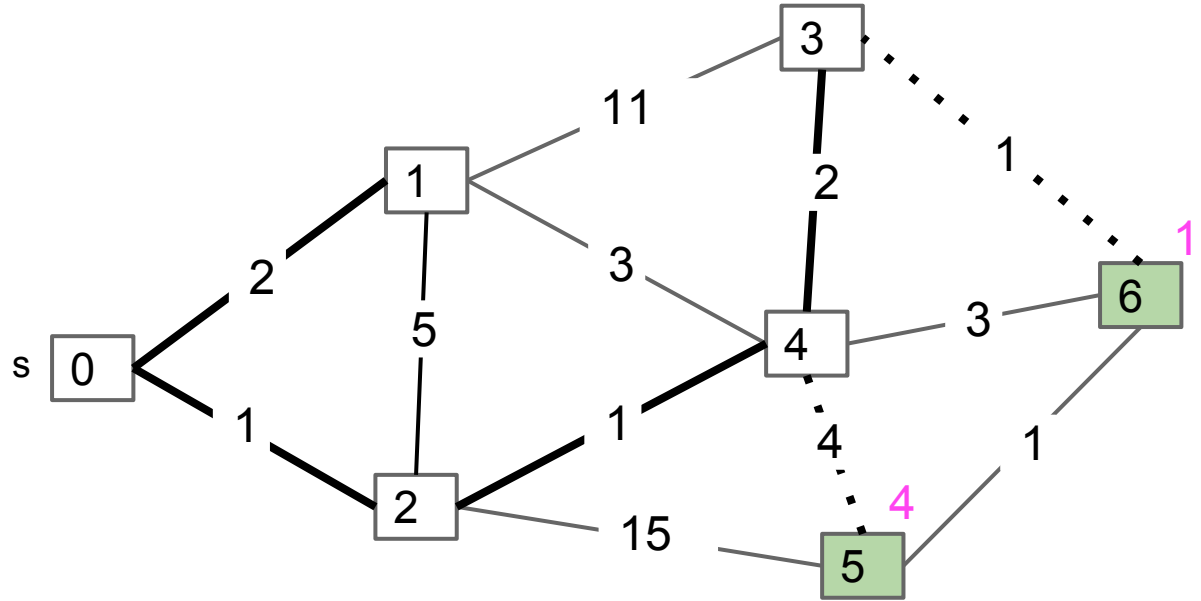


Fringe: [(6: 1), (5: 4)]

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0		-
1		0
2		0
3		4
4		2
5	4	4
6	1	3



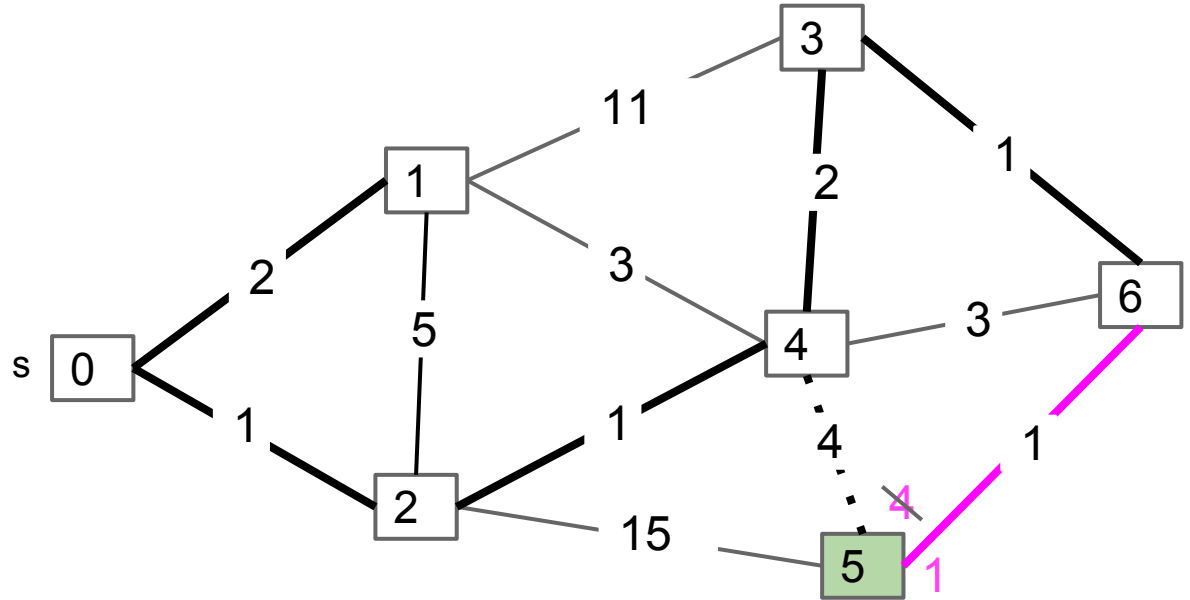
Fringe: [(6: 1), (5: 4)]

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0		-
1		0
2		0
3		4
4		2
5	1	6
6		3

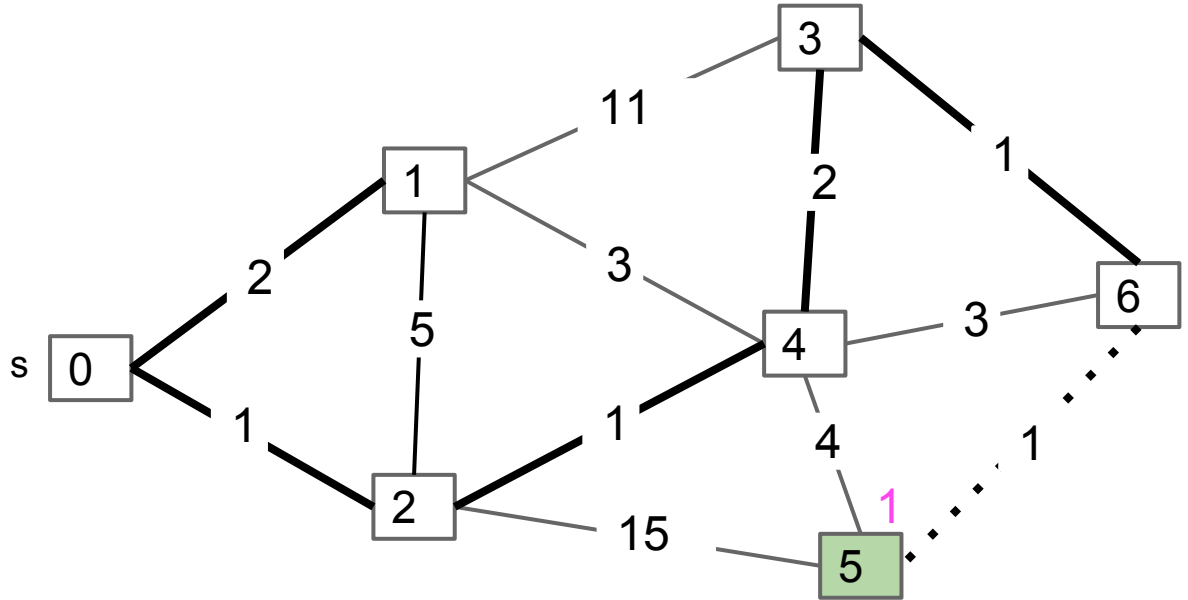
Fringe: [(5: 1)]



Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0		-
1		0
2		0
3		4
4		2
5	1	6
6		3

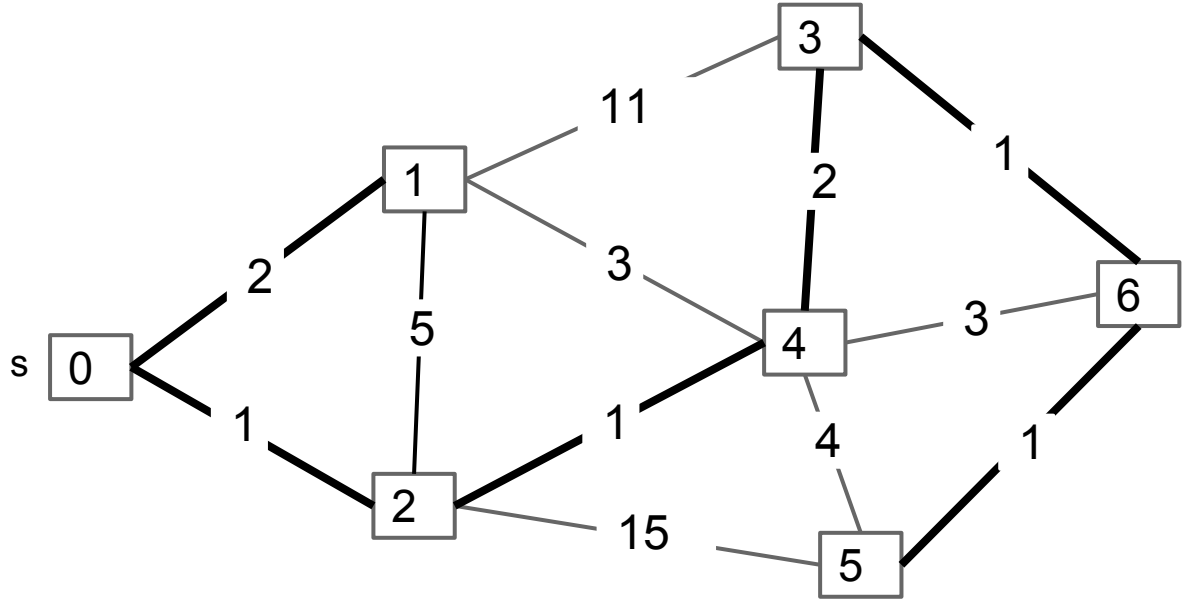


Fringe: [(5: 1)]

Prim's Demo

Insert all vertices into fringe PQ, storing vertices in order of distance from tree.
Repeat: Remove (closest) vertex v from PQ, and relax all edges pointing from v .

#	distTo	edgeTo
0	-	-
1	0	
2	0	
3	4	
4	2	
5	6	
6	3	



Fringe: []