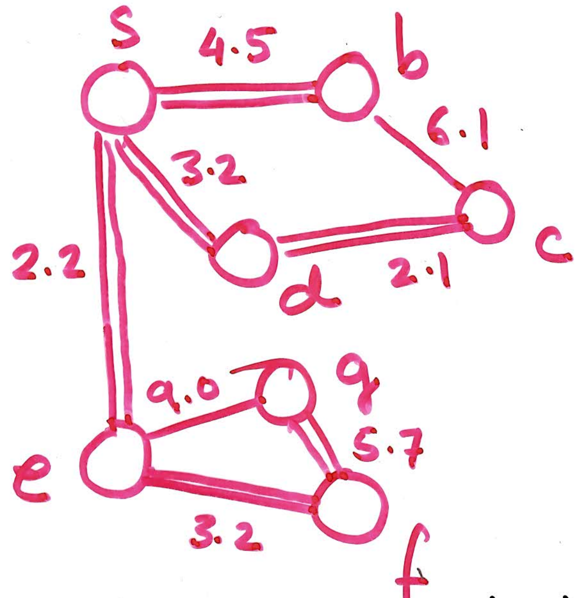
Single Source Shortest Paths

Given a graph (connected) $G=(V,E)$ with weights on edges and source vertex $s \in V$, find the shortest path from s to any other vertex $v \in V$.

Ex.



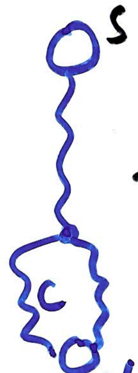
$\text{Length}(s,b,c) = 10.6$

$\text{length}(s,d,c) = 5.3$

$\delta(s,c) = 5.3$

The graph produced by the shortest paths form a <u>spanning tree</u>.

Why no cycle?



There are two paths from s to v. Assuming only one path to v is created if there are two or more paths of equal lengths, c cannot be present.

Dijkstra' Algorithm.

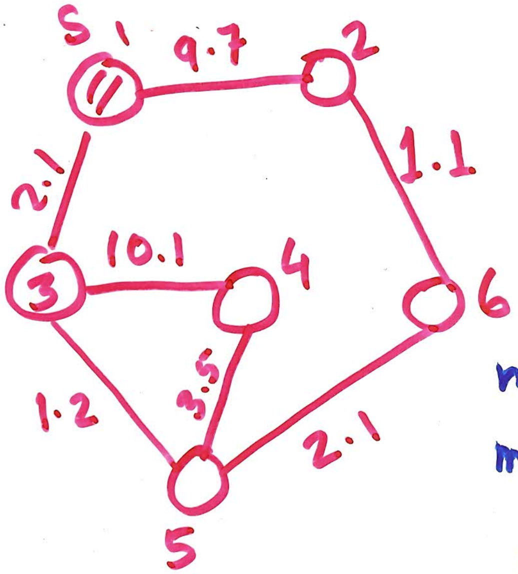
Starting from the source s, we grow a shortest path tree and gradually expand it. For all vertices v that are not yet in the tree. We maintain a p-value that reflects the length of the path from s to v using only vertices in the tree except v.

The Algorithm.

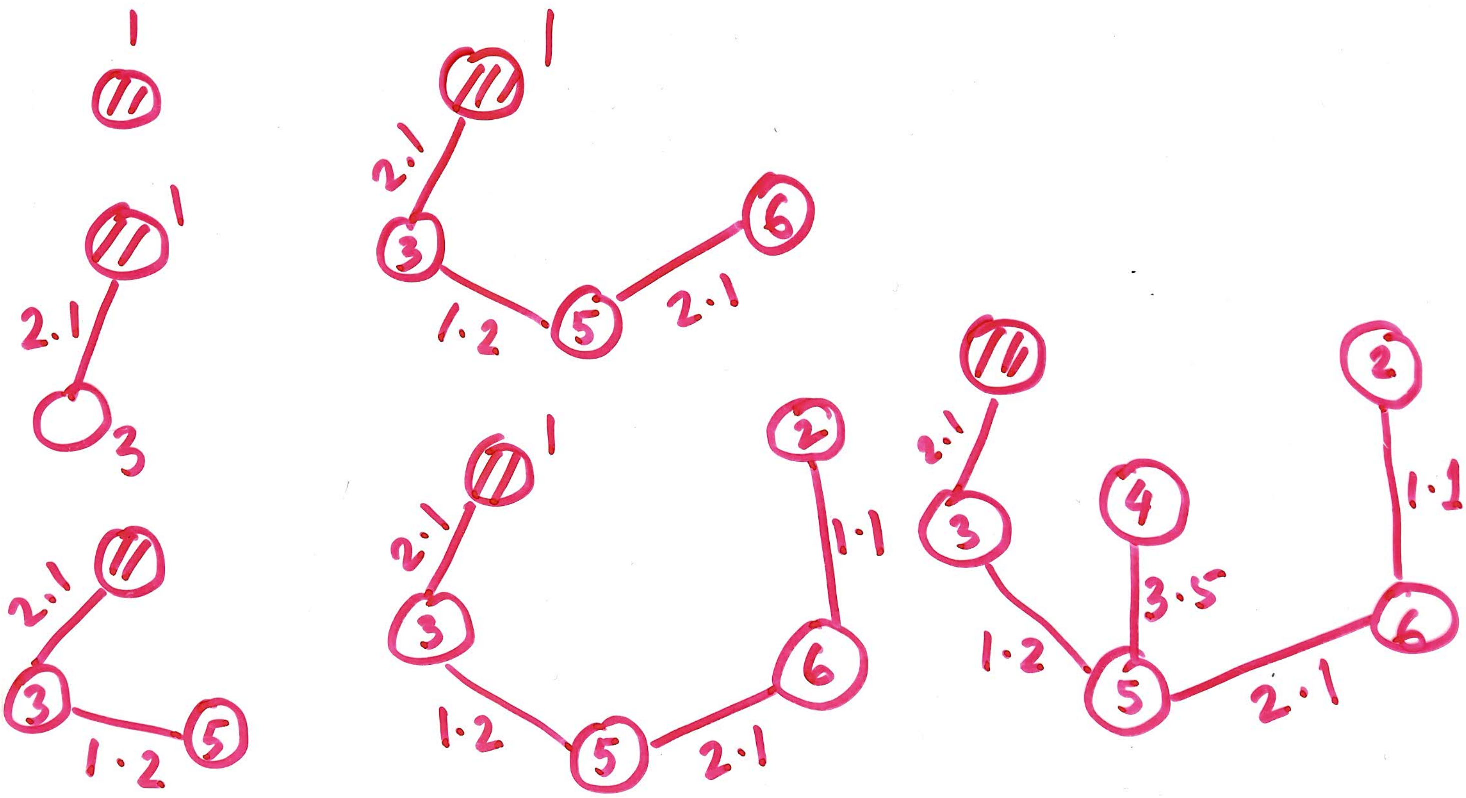
```
PQ := ∅; for i:=1 to n do
        if i≠k then V[i].p := ∞
           else V[i].p := 0; V[i].π := nil;
        endif
        add i with priority V[i].p to PQ
        endfor

While PQ ≠ ∅ do
   i := Extractmin(PQ); t := V[i].adj;
   While t ≠ nil do
      j := t.v;
      if j ∈ PQ and V[i].p + ω(i,j) ≤ V[j].p then
            Δ := V[j].p - (V[i].p + ω(i,j));
            Decrease_key (PQ, j, Δ);
            V[j].π := i;
      endif
   endwhile
endwhile
```

Example.



	1	2	3	4	5	6
	0	∞	∞	∞	∞	∞
min 0		9.7	2.1	∞	∞	∞
min 2.1		9.7	~~2.1~~	12.2	3.3	∞
min 3.3		9.7		6.8	~~3.3~~	5.4
min 5.4		6.5		6.8		~~5.4~~
min 6.5		~~6.5~~		6.8		
min 6.8				~~6.8~~		



Time Complexity:

If PQ is an array : $O(n^2)$

If PQ is and ordinary heap: $O((n+m)\log n)$

If PQ is a Fibonacci heap: $O(m + n\log n)$

n Extract_min
m Decrease-key

Correctness: Let S be the set of vertices marked so far and r be the vertex being included into S.

Inductive Hypothesis:

(i) For any vertex $i \in S$, $V[i].p = \delta(s, i)$

(ii) For any vertex $i \notin S$, $V[i].p$ is the length of the shortest path to i that goes through only the vertices in S except i itself.

Base Case $S = \{s\}$, $V[s].p = 0$ and $V[i].p = w(s, i)$ when i is adjacent to s and $V[i].p = \infty$ for all other i.

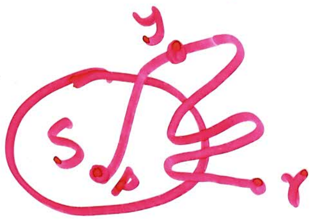
(i) & (ii) are satisfied trivially.

Inductive step.

V[r].p is minimum among all vertices that do not belong to S.

Claim shortest path to r lies entirely in S except the vertex r.

Proof. Suppose on the contrary, it lies outside S also, y be the first vertex on the path outside S. In that case



V[y].p < V[r].p, ~~a contradict~~ a contradiction

By [ii] V[r].p represents the length of the shortest path to r. This proves (i).

Now we prove (ii)

The p-values that are affected by including r into S are those vertices which are adjacent to r.

Let y be such a vertex.
Let the shortest path from s to y without including is P_1.

<u>Claim</u> Shortest path to y including r is the path $(s \to r) + (r, y)$, where $(s \to r)$ is the shortest path from s to r that lies entirely in S except r.

Suppose it is not. Let b be such a path. Consider the last vertex z on this path that is in S. The path from s to z through r is shorter than s to z without r, a contradiction to (i).

Thus $(s \to r) + (r, y)$ is the shortest path to y including r. $V[y].p$ is updated as $V[y].p = \min(V[y].p, V[r].p + w(r, y))$ maintaining (ii).