Example:

Initialize:
Assume root node is a in above example:

- a.key = 0
- a.π = NIL

All other vertices u have u.key=∞ and u.π=NIL.

Q = abcdefghi

```
PRIM(G, w, r)
Q = ∅
for each u ∈ G.V
   u.key = ∞
   u.π = NIL
   INSERT(Q, u)
DECREASE-KEY(Q, r, 0)  // r.key = 0
while Q ≠ ∅
   u = EXTRACT-MIN(Q)
   for each v ∈ G.Adj[u]
      if v ∈ Q and w(u, v) < v.key
         v.π = u
         DECREASE-KEY(Q, v, w(u, v))
```
**Extract #1:**
Extract Min removes a since a.key = 0 is smallest
Q = bcdefghi

**Update:**
b, c in Adj[a], still in Q and edge weights for (a,b) and (a,c) both less than the b.key and c.key, respectively, so replace keys for b and c and set parents for both b and c to a:

b.key = 10
b.π = a
c.key = 12
c.π = a

**Min-Heapify to get:** Q = bcdefghi

**Extract #2:**
Extract b
Q = cdefghi

**Update:**
a, c, d in Adj[b], and a not in heap so ignore. For vertex c, weight w(b,c)<c.key, so update c.key to 9, set parent to b. For d, weight w(b,d)<d.key=∞ so update as well:

c.key = 9
c.π = b
d.key=8
d.π = b

**Min-Heapify to get:** Q = dcefghi

**Extract #3:**
Extract d:
Q=cefghi

**Update:**
b, e, h, g in adj[b], b not in heap so ignore. For e, update since w(d,c)< e.key=∞. For h, update and for g, update:

e.key = 7
e.π = d
h.key = 5
h.π = d
g.key = 8
g.π = d

**Min-Heapify to get:** Q= hegcfi
**Extract #4:**
Extract h:
Q = egcfi

**Update:**
f, d, g, i in adj[h], d not in heap so ignore. For f, update, and for g, don’t update since g.key=8 which is not less than w(g,h)=9. For i, update.

f.key = 6,
f.π = h
i.key = 11,
i.π = h

Min-Heapify to get: Q = fegci

**Extract #5:**
Extract f:
Q = egci

**Update:**
c, e, and h in adj[f], h not in heap so ignore. For e, update since w(f,e)=3 < e.key = 7. For c, update.

e.key = 3
e.π = f
c.key = 1
c.π = f

Min-Heapify to get: Q = cegi

**Extract #6:**
Extract c
Q = egi

**Update:**
a, b, e, f in adj[c] but only e in heap. Since e.key = 3, don’t update (only update if edge weight is strictly less than the current key)

Min-Heapify to get: Q = egi

**Extract #7:**
Extract e
Q = gi

**Update:**
c, f, d in adj[e] but none on heap so done.

Min-Heapify to get Q = gi
**Extract #8:**
Extract g
Q=i

**Update:**
d, h, and i in adj[g], and only i on heap. Update i since w(g,i)=2 < i.key=∞.

i.key = 2
i.π = g

Min-Heapify to get Q=i

**Extract #9:**
Extract i
Q=empty

**Update:**
g, h in adj[i], but none on heap so done.

Q is empty, halt.

---

Notice that we can then construct the MSP by including the edge from each vertex to it’s parent node as is done in the above example.