PRIM's MST algorithm

- Start with an arbitrary vertex r. Grow MST by repeatedly adding the smallest edge connecting a vertex in the tree with a vertex not in the tree
- To find the smallest edge we use a priority queue containing the *vertices* not in the tree yet:
  - The key/priority $d[v]$ of a vertex v is the weight of the smallest edge connecting v to the tree (implementation note: the priority of a vertex will be stored in an array $d[v]$ and also in the priority queue $(v, d[v])$; in order to be able to decrease the prioriy of a vertex we store a pointer to v's location in the priority queue)
  - For each vertex v we store the edge that connects it to the tree; we call the other vertex of this edge by $pred(v)$

PRIM(G)
// initialize
Pick arbitrary vertex r and set $d[r] = 0$, PQ.Insert$(r, 0)$, $pred(r) = NULL$
For each vertex $u \in V (u \neq r)$: $d[u] = \infty$, PQ.Insert(u, ∞)

while PQ not empty
    u = PQ.Delete-min() // u is vertex closest to the tree
    For each adjacent edge (u, v)
        IF v in PQ and $w_{uv} < d[v]$
            PQ.Decrease-Key(v, w_{uv})
            pred$[v] = u$
Output the edges $(u, pred(u))$ as the MST.

Analysis: $O(|E| \lg |V|)$

Kruskal's MST algorithm

KRUSKAL(G)
// initialize
For each vertex $v \in V$: Make-Set(v)
Sort edges of E in increasing order by weight

for each edge $e = (u, v)$ in order of weight
    if Find-Set$(u) \neq$ Find-Set(v)
        output edge e as part of MST
        Union-Set(u, v)

Analysis: $O(|E| \lg |V|)$