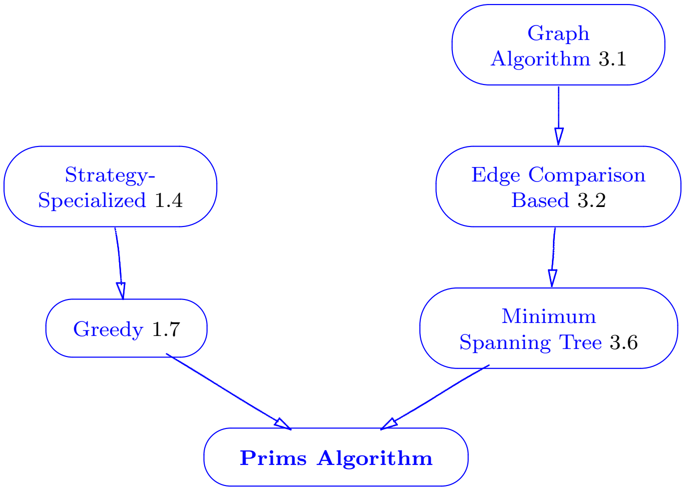
3.3 Prim's Minimum Spanning Tree

Section authors: Joel Branch, Matt Freeman,
Steve Halaka



Refinement of: Greedy (§1.7) Minimum Spanning Tree (§3.6). Prim's algorithm is a greedy algorithm for solving the Minimum spanning tree problem. It is closely related to Dijkstra's (§3.16) single source shortest path algorithm.

Prototype: `template` $\langle$*class VertexListGraph, class DijkstraVisitor, class PredecessorMap, class Dis*

```
void prim_minimum_spanning_tree(
   const VertexListGraph& g,
   vertex_descriptor s, PredecessorMap pred,
   DistanceMap distance, WeightMap weight,
   IndexMap index_map)
```

Input: Directed or undirected graph G, a model of VertexListGraph, with edge set E and vertex set V.
Start node s.

WeightMap, which maps each $e \in E$ to real values.
Empty PredecessorMap, which will map each $v_1 \in V$ to another $v_2 \in V$, where v_2 corresponds to the parent of v_1 in the minimum spanning tree.
IndexMap, which maps each $v \in V$ to an integer in the range $[0,$ num_vertices(g)).
Empty DistanceMap, which will map each $v \in V$ to its shortest path weight from source vertex s in the minimum spanning tree. [1]

Output: Modified PredecessorMap, which maps each $v_1 \in V$ to another $v_2 \in V$, where v_2 corresponds to the parent of v_1 in the minimum spanning tree, if one exists.
A Modified DistanceMap, which maps each $v \in V$ to its shortest path weight from source vertex s in the minimum spanning tree.

Effects: No effects on the input.

Asymptotic complexity: Let $E =$ number edges in the graph and let $V =$ number of vertices in the graph.

- Best case (sparse graph where $E \lessapprox V$): $O(V \log_2 V)$
- Average case: $O(E \log_2 V)$
- Worst case (dense graph where $E \approxeq V^2$): $O(V^2 \log_2 V)$

Operation Counts in the Average Case:

V	E	Assignments	Iterator	Integer	Compare
100	100	414	514	400	102
100	200	2914	1800	1125	980
100	400	5150	3288	2287	1894
100	800	6946	5710	4301	2676
100	1600	9772	10510	8301	3928
100	3200	14648	20110	16301	6263
100	6400	23270	39310	32301	10716

Value comparisons:	0.27$E \log_2 V$ + 0.08$V \log_2 V$
Value assignments:	0.6$E \log_2 V$ + 0.31$V \log_2 V$
Iterator operations:	0.93$E \log_2 V$ + 0.39$V \log_2 V$
Integer operations:	0.76$E \log_2 V$ + 0.31$V \log_2 V$

V	E	Assignments	Iterator	Integer	Compare
100	100	414	514	400	102
200	100	964	1104	836	232
400	100	1614	2014	1600	402
800	100	3214	4014	3200	802
1600	100	6414	8014	6400	1602
3200	100	12848	16034	12808	3209
6400	100	25614	32014	25600	6402

Algorithm Animation: An animation for Prim's MST algorithm can be found both here.