Prim's Algorithm

Definition:

Prim's algorithm, also known as Jarnik's algorithm, is a greedy minimum spanning tree algorithm that takes a graph as input and finds the subset of the edges of that graph that:

- form a tree that includes every vertex.
- has the minimum sum of weights among all the trees that can be formed from the graph.

Prim's shares a lot of similarity with the shortest path first algorithms.
In contrast to Kruskal's, **the nodes are treated as a single tree** that we continuously add to.

Pseudocode

In the pseudocode below we initialize our tree *T* to null, create a list of visited vertices *U*, V-U is the list of vertices that are unvisited.
We one by one move the vertices from set V-U to U by **connecting the least weight edge**.

```
function prims(V, E):
T = ∅
U = { 1 }
while U != V:
        let (u, v) be the lowest cost edge such that u ∈ U and v ∈ V - U
        T = T ∪ {(u, v)}
        U = U ∪ {v}
```

Source

Algorithm Description

Starting from a vertex, continously add edges with the lowest weight until reaching the goal stated above.
There are three steps for implementing this pattern.

1. Start a new tree by choosing a random starting vertex.
2. Find all the edges that connect the tree to new vertices,
3. Keep repeating step 2 until we get a minimum spanning tree.

Essentially you are initializing a tree *T* and **repeatedly adding safe edges** until you result in a minimum spanning tree.

Time Complexity

The worst case time complexity of this algorithm is **O(E log V)**.