Prim's Algorithm

Kruskals Remarks

1. Joseph Kruskal published this algorithm in 1956, this was two years after he finished his PHD at princeton. He is known for work in CS, combinatorics and stats.
2. The cost of an edge is sometimes called, the ***weight*** of that edge.
3. Kruskals algorithm starts w/ a disconnected graph $(V, \emptyset)$ and adds edges until the graph becomes connected and a tree, thus a spanning tree as $V\prime = V$. In other words, after the last addition the graph is disconnected.

Prim's Algorithm history

TASK Describe another algorithm for constructing minimal spanning trees, which is denoted by the fact that each step of the algorithm, the subgraph is a tree. The algorithm is called Prim's Algorithm.

Vojtéch Jarník first discovered and published the algorithm in 1930. Robert Prim subsequently re-discovered it and published it in 1957. it was once again rediscovered by Edger Dijkstra in 1959.

Moral of the story

The idea behind this algorithm is very natural. We apply procedure **2** for constructing a spanning tree that we discussed using the same queue of edges ordered by cost as in Kruskals algoritm. The result at each step is a tree and at the end we get a minimal spanning tree.

Prim's Algorithm Description

let (V,E) be a connected graph w/ an associated cost function $c : E \to R$

1. Start by choosing some vertex $v \epsilon V$, The starting subgraph is $(v, \emptyset)$
2. List all edges in E in a queue so that the cost of the edges is non-decreasing in the queue i.e if $e, e\prime \epsilon E$ and $c(e) < c(e\prime)$, then e precedes e` in the queue.
3. We identify the first edge in the queue which has one vertex included in the current subgraph and the other vertex not included in the subgraph. We add that edge to the current subgraph as well as the vertex that was no included. Since the subgraph with which we started was a tree, the resulting subgraph is a tree (we added one vertex and one edge). Continue the process until it is not possible to proceed any further. i.e we have added all vertices in V.
4. The fact that at each stage we have a tree, and at the end that tree contains all vertices in V guarantees that Prim's algorithm yields a spanning tree. The fact that we chose what edge to add next by following the queue of edges ordered by cost guarantees that what we obtain is a minimal spanning tree of the original undirected graph (V,E).

Example

Consider the following queue and graph :

See the sample example here

We have used all the vertices of the original graph, so the algorithm ends at this step. Prim's algorithm produces the same tree as Kruskal's, given the same queue.

Remarks

1. Just like Kruskal's algorithm, Prim's algorithm produces a unique minimal spanning tree if no two edges have the same cost. If there are edges with the same cost, reshuffling them yields different queues that in turn yield different spanning trees.
2. We make a choice as to which vertex kickstarts Prim's algorithm. Different choices yield different trees. at the intermediate stages of the algorithm.

- **DEF** The minimal spanning tree yeilded by prims algorithm is called the prim *spanning tree*.
- **DEF** Let (V_i, E_i) be the subgraph at the end of step i of Prims's algorithm. All vertices in V_i are called visited vertices. If

(V, E) is the oriiginal connected graph on which Prim's algorithm is being applied all vertices in $V \not V_i$ one called the unvisited vertices

Applications of minimal Spanning Trees

- Design of networks such as
    - Computer Networks,
    - Transportation networks
    - Telecommunication
    - water supply networks
    - electrical grids
- Computer minimal spanning trees appears as a subroutine in algorithms
    - NP-Hand problems
    - Traveling salesman problem
- Minimal spanning trees can be used to describe financial markets, in particular how stocks are calculates
- Various other problems in Computer Science and engineering.