Algorithms Design and Analysis [ETCS-301]

Dr. A K Yadav
Amity School of Engineering and Technology
(affiliated to GGSIPU, Delhi)
akyadav1@amity.edu
akyadav@akyadav.in
www.akyadav.in
+91 9911375598

October 21, 2019

Prim's algorithm I

- Prim's algorithm has the property that the edges in the set A always form a single tree unlike Kruskal.
- We starts taking any node r as root and grows till the tree spans all the vertices in V
- In each step a new minimum edge vertex is added into A
- The algorithm uses greediness, as each step it adds to the tree an edge that contributes the minimum weight to the tree.
- We use min-priority queue Q based on key to store all vertices that are not the part of the MST till now.
- For each vertex v, the value $key[v]$ will be the minimum weight of the edge connected v with all other nodes of the graph.
- if any node is not connected with node v then $key[v] = \infty$
- $key[r] = 0$ as r is the root of the tree

Prim's algorithm II

- As we are expending tree by taking the node with minimum weight edge at every step so the tree will be MST at every step.
- $\pi[v]$ is the predecessor of v. For example if (u, v) is the edge of the tree then $\pi[v] = u$
- $adj[u] = v$ if there is an edge from u to v that is (u, v)

Algorithm for Prim I

MST-PRIM(G,w,r)

1. $A = \emptyset$
2. $Q = \emptyset$
3. for each $u \in V$
4. $\quad key[u] = \infty$
5. $\quad \pi[u] = NIL$
6. $\quad Insert(Q, u)$
7. $key[r] = 0$
8. while $Q \neq \emptyset$
9. $\quad u = Extract(Q)$
10. $\quad$ for each $v \in adj[u]$
11. $\quad\quad$ if $v \in Q$ and $w(u, v) < key[v]$
12. $\quad\quad\quad \pi[v] = u$

Algorithm for Prim II

13. $key[v] = w(u, v)$
14. $A \cup \{(u, v)\}$
15. return

Time complexity of the algorithm is $O(V \lg V + E \lg V)$ or $O(E \lg V)$ using Min-heap and $O(E + V \lg V)$ using Fibonacci heap.

Thank you

Please send your feedback or any queries to akyadav1@amity.edu, akyadav@akyadav.in or contact me on +91 9911375598