Readings Ch. 23

Self test Ex. 23.1-1, 23.2-2

Lecture 19

Minimum Spanning Trees

Let $G = (V, E)$ be a connected undirected graph with edge weights $w(e)$ for edge e in E.

A tree is an acylcic graph.

A spanning tree is a set of edges that does not contain a cycle and every vertex v in V is an endpoint of at least one edge.

A minimum cost spanning tree (MCST) is a spanning tree A such that the sum of the weights is minimized for all possible spanning trees B.

$$w(A) = \sum_{e \in A} w(w) \leq w(B)$$

Prim's Algorithm

Define 2 arrays p such that $p[u]$ contains vertex v such that (u, v) is in E and $w(u, v)$ is minimized for all v in T adjacent to u. Priority such that priority$[u]$ contins $w(u, p[u])$.

```
Prim-MST(G=(V,E) w: E → R)
  T = \{\}
  initialize priority queue Q
  for all v in V:
    priority[v] = ∞
    p[v] = NIL
    Insert(Qv)
    pick arbitrary vertex s in v
    priority[s] = 0
  while (not isEmpty(Q)) do:
    u = extract-min(Q)
    if p[u] ! = NIL
      T = T ∪{(p[u], u)}
    for each v in adjacency-list[u]:
      if v in Q and w(u,v) < priority[v]:
        decrease-priority(v, w(u,v))
        p[v] = u
```