Algorithm HW #3

11/8/2017
Due: 11/8 4:50pm
장소: 302동 313-2 최적화연구실

1. G = (V, E)가 weighted digraph이고 negative-weight cycle은 없다. T는 vertex *v*를 root로 삼아 G로부터 만들어진 임의의 minimum spanning tree이다. 이 spanning tree T가 *v*로부터 각 vertex에 이르는 shortest path들만 포함하고 있는지 여부를 확인하는 linear-time algorithm을 제시하라. 이 알고리즘은 Yes 또는 No의 대답만 할 수 있으면 된다. 제시한 알고리즘이 linear time에 수행된다는 것을 설명하라. 또한 당신이 제시한 알고리즘이 제대로 작동한다는 것을 수학적 귀납법을 이용하여 증명하라.

2. Edge들의 weight이 모두 다른 그래프는 단 하나의 minimum spanning tree를 가짐을 증명하라.

3. Bellman-Ford algorithm은 negative-weight cycle이 없는 graph G = (V, E)에서 임의의 starting vertex s로부터 다른 vertex들로 가는 shortest path를 구하는 알고리즘이다. 이 알고리즘을 s로부터 각 vertex에 이르는 가장 짧은 두 개의 shortest path를 구하는 알고리즘으로 바꾸어 보아라. 이 알고리즘의 running time은 어떻게 되는가? 또한 이 알고리즘의 결과물을 이용하여 vertex s에서 임의의 vertex *t*로 가는 가장 짧은 path 두 개를 print out 하는 알고리즘도 제시하라.

4. Digraph $G = (V,E)$의 transitive closure (|V|×|V| matrix)를 만드는 작업을 색다르게 해본다. 초기에 G의 edge를 모두 제거한 다음 (즉, matrix의 element들이 모두 0), edge들을 하나씩 더해 가면서 transitive closure를 update하는 방식으로 만들어 보려 한다.

   4.1 임의의 edge가 G에 더해질 때 transitive closure는 $O(|V|^2)$ time에 update될 수 있음을 보여라.
   4.2 이런 방식으로 모든 edge를 차례로 더하면 되므로, 위 2.1에 의하여 digraph G의 transitive closure를 구하는 데는 $O(|E||V|^2)$ 시간이면 된다. 이를 Warshall이 사용한 시간과 같은 $O(|V|^3)$ 시간으로 개선할 수 있는 방법이 있다. 이를 찾고 그 타당성을 설명하라.

5. $G = (V, E)$는 undirected weighted graph이고, T는 G의 한 minimum spanning tree이다. 이제 G의 한 edge e의 weight이 바뀌었다고 하자. T가 여전히 minimum spanning tree인지 아닌지 알아내는 알고리즘을 제안하고 제안한 알고리즘의 asymptotic running time을 밝히라.

   Note: e가 T에 속해 있는 경우와 그렇지 않은 경우로 나누어 생각하라.
   알고리즘은 말로 설명하는 것으로 충분함 (중요한 것은 의미 전달).