

자료구조설계

- 실습05 : Prim 알고리즘 구현 -

제 출 일	2016.10.23
분 반	02
담당교수	박정희
학 과	컴퓨터공학과
학 번	201302423
이 름	신종욱

1. 구현 내용

```
import java.util.*;

public class PrimClass {
    ArrayList<Node> T;//연결된 정점과 비용을 저장하는 리스트
    HashSet<String> TV;//현재 연결되고있는 정점들을 저장하는 집합,방문 유무도 판단가능
    String[] V;//정점 정보를 저장하는 배열
    int[][] E;// 정점들의 연결가중치를 저장하기위한 배열
    int cost;//총 비용 계산을 위한 정수값

    public void make(String[] A, int[][] B) {
        V = A;
        E = B;
        T = new ArrayList<Node>();
        TV = new HashSet<String>();
    }

    public void prim(String A) {
        int a = -1, b = -1;// T와 TV에 저장할걸 기억하기위해 쓰는 정수
        TV.add(A);//일단 시작하는 것은 무조건 연결될 예정이기 때문에 미리 연결한다
        int m = Integer.MAX_VALUE;

        while (T.size()+1 != V.length) {
            int mincost=m;
            for (int k = 0; k < V.length; k++) {
                for (int j = 0; j < V.length; j++)
                    if (TV.contains(V[k])&&!TV.contains(V[j])&&E[k][j] < mincost) {
                        mincost = E[k][j];
                        a = k;
                        b = j;//집합에 포함된 문자열에서 포함안된 문자열로 뺄어가는 간선중
                        //가장 가중치 작은걸 찾는다
                    }//최종으로 찾은값은 a,b로 인덱스들이 저장되어있다.
            }
            System.out.println("선택한 간선 : (" + V[a] + ", " + V[b] + ", " + mincost+");");
            cost += mincost;
            Node temp = new Node(a, b, mincost);
            T.add(temp);//배열들과 비용을 저장해서 T에 더해준다
            TV.add(V[b]);//최종선택되어 연결된부분을 집합 V에 더해준다.
        }
    }

    class Node {
        int in;
        int out;
        int cost;
        // 가중치 배열의 행렬위치를 기억해서 정점의 정보를 저장하기위한 정수
        Node(int in, int out, int cost) {//생성자 선언
            this.in = in;
            this.out = out;
            this.cost = cost;
        }
    }
}
```

Prim 메소드 추가 설명 :

Edge를 선택해서 연결한 T를구현 하기위해서 Node를 선언하여서 인덱스값 2개와 비용을 저장하도록 선언하고 T를 ArrayList로 만들어서 관리하기로 하였다.

연결된 정점 집합을 나타내는 TV를 구현하기위해 Hashset을 썼다.

일단 prim은 정점중 하나를 받는데 어차피 그정점을 포함해서 시작해야하기 때문에 바로 TV집합에 넣어준다.

T.size()+1인 이유는 MST는 항상 정점수-1 만큼의 Edge로 구성되기 때문이다.

모든 가중치 배열을 돌면서 TV에는 속하는 정점에서 TV에 속하지 않는 정점중 최소 가중치를 찾는다.

최종적으로 찾은 값을 T와 TV에 저장한다.

(사실 이구현에서 TV를 사용하지 않아도 출력에는 영향을 주지않지만 최종값을 확인을 위해 사용하여야 한다)

2. 실행 결과

<terminated> TestPrimClass [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (2016. 10. 21. 오후 11:20:19)

선택한 간선 : (A, F, 1)
선택한 간선 : (A, G, 2)
선택한 간선 : (G, C, 1)
선택한 간선 : (C, B, 2)
선택한 간선 : (G, D, 2)
선택한 간선 : (D, E, 1)
비용합계 : 9