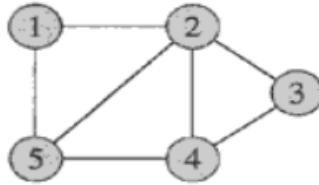


## 제12회 연습문제

- 하나의 그래프에 대한 인접행렬이 입력으로 주어질 때 그 그래프에 대한 인접리스트를 만드는 프로그램을 작성하라. 입력으로 먼저 정점의 개수  $n$ 이 주어지고, 이어서  $n \times n$ 크기의 행렬이 주어진다. 인접리스트를 만든 후 각 정점에 대해서 인접한 정점들을 출력하라. 정점의 번호는 1번에서  $n$ 번까지이다. 아래의 입출력 예는 다음 그림의 그래프의 경우이다. (출력만 해서는 안되며 실제로 인접리스트를 만들어야 한다.)



### 입력 예

```
5
0 1 0 0 1
1 0 1 1 1
0 1 0 1 0
0 1 1 0 1
1 1 0 1 0
```

### 출력 예

```
1: 2 5
2: 3 4 1 5
3: 2 4
4: 2 3 5
5: 1 2 4
```

- 캘리포니아 주의 모든 도시(?)들의 이름과 위도 및 경도가 명시된 샘플 데이터파일이 주어진다. 이 파일에 등장하는 모든 도시들을 정점으로 하고 두 도시간의 거리가 10km 이하이면 에지로 연결하는 그래프의 인접리스트를 구성한 후, 사용자가 도시의 이름과 하나의 정수  $k$ 를 입력하면 그 도시로 부터  $k$  홉(hop) 이내에 있는 모든 도시들의 이름을 출력하는 프로그램을 작성하라. 여기서  $k$ 홉 이내라는 말은 그래프에서 두 노드를 연결하는 경로의 길이(에지의 개수)가  $k$  이하라는 의미이다. 인접리스트를 구성하기 위해서는 도시의 이름에 일련번호를 부여해야 한다. 이 번호는 데이터 파일에 나오는 순서대로 부여하라. 또한 도시 이름에 부여된 일련 번호를 빠르게 검색할 수 있어야 한다. 이를 위해서 이진검색트리를 이용하라. 즉 도시이름을 key로, 일련번호를 value로 해서 <도시이름, 일련번호> 쌍을 이진검색트리에 저장한 후 도시이름으로 검색하면 일련번호를 알 수 있다. 데이터 파일에서 필드들은 tab문자로 분리되어 있고, 첫 줄은 테이블 헤더이므로 무시하면 된다. 동일한 도시 이름이 중복해서 등장하기도 한다. 이 경우 이름 뒤에 다시 1, 2,... 식으로 번호를 추가하여 구분한다. 다음은 위도 경도 값으로 부터 거리를 계산해주는 함수이다. 도시간의 거리를 계산할 때 이 함수를 이용하라. 이 함수가 반환하는 거리의 단위는 m이다.

```
public double calDistance(double lat1, double lon1, double lat2, double lon2)
{
    double theta, dist;
    theta = lon1 - lon2;

    dist = Math.sin(deg2rad(lat1)) * Math.sin(deg2rad(lat2))
           + Math.cos(deg2rad(lat1))
             * Math.cos(deg2rad(lat2)) * Math.cos(deg2rad(theta));
    dist = Math.acos(dist);
    dist = rad2deg(dist);

    dist = dist * 60 * 1.1515;
    dist = dist * 1.609344; // 단위 mile 에서 km 변환.
    dist = dist * 1000.0;   // 단위 km 에서 m 로 변환
    return dist;
}
```

```
// 주어진 도(degree) 값을 라디언으로 변환
private double deg2rad(double deg){
    return (double)(deg * Math.PI / (double)180d);
}

// 주어진 라디언(radian) 값을 도(degree) 값으로 변환
private double rad2deg(double rad){
    return (double)(rad * (double)180d / Math.PI);
}
```