알고리즘

- HW11: BFS,DFS -

제 출 일	2017년 12월 12일
분 반	02반
담당교수	공은배
학 과	컴퓨터공학과
학 번	201302423
이 름	신종욱

ㄱ.BFS

1. 해결 방법

```
    boolean[][] bool = {
            {true,true,false,false,true,false,false,false},//r
            {true,true,false,false,false,true,false,false},//s
            {false,false,true,true,false,true,true,false},//t
            {false,false,true,true,false,false,true,true},//u
            {true,false,false,false,true,false,false,false},//v
            {false,true,true,false,false,true,true,false},//w
            {false,false,true,true,false,true,true,true},//x
            {false,false,false,true,false,false,true,true},//y
            };//정점들의 연결상태를 선언
    int M = Integer.MAX_VALUE;
    node[] W = new node[bool.length];
    for(int i=0;i<W.length;i++) {
            W[i]=new node(M, -1,"white");
    }//각각 노드들의 상태를 지정
    int start =1;//첫노드 선택 1번째 인덱스 선택
    W[start]=new node(0,-1,"gray");//root의 부모의 경우에는 -1로 표현
    Queue<node> queue = new LinkedList<node>();//노드관리 큐
    Queue<Integer> queuewhatnum = new LinkedList<Integer>();//접근 한노드 인덱스 저장하기위한 큐
    queue.add(W[start]);
    ////////////////초기화
    while(!queue.isEmpty()) {//큐가 빌때까지
        node u = queue.poll();//큐의 맨앞을 꺼내서 저장
        for(int k =0;k<bool.length;k++) {
            if(bool[start][k]==true)
                if(W[k].color.equals("white")) {
                    W[k].color="gray";
                    W[k].distan=u.distan+1;
                    W[k].parent=start;
                    queue.offer(W[k]);//큐뒤에 추가
                    queuewhatnum.offer(k);//큐 맨 뒤에 추가
                }
        }
        u.color="black";
        if(!queue.isEmpty()) {
        start=queuewhatnum.poll();}
    }
    for(int i=0;i<W.length;i++) {
    System.out.print("인덱스 번호 : "+i+" 시작노드에서 거리 :"+W[i].distan+" ");
    System.out.println("부모노드 : "+W[i].parent);
    }
}
```

노드의 연결상태를 boolean배열로 선언하고 각 노드들의 상태를 초기화한다 첫 번째 시작노드를 1로 선택하고 큐도 선언후 1을 넣어 초기화한다.
그 후 큐가 빌때까지 큐에서 꺼낸 노드가 연결된 노드들을 방문하면서 큐에 넣으면서 방문했다는 색표시와 거리를 갱신하면서 반복문을 실행한다.

아래는 노드객체 선언문이다

```
static class node {
    int     distan;
    int     parent;
    String  color;
     public node(int x,int y,String z){
         this.distan=x;
         this.parent=y;
         this.color = z;
     }

}
```

루트로 부터의 거리,부모노드,방문 여부를 저장한다

2.실행결과

ㄱ.DFS

1. 해결 방법

```
13
14          node[] W = new node[bool.length];
15          for(int i=0;i<W.length;i++) {
16                  W[i]=new node(-1,-1, -1,"white");//시간 초기화는 -1,루트노드는 부모는 -1로 표시
17          }//각각 노드들의 초기 상태를 지정
18          time =0;
19          for(int i=0;i<W.length;i++) {
20              if(W[i].color.equals("white")) {
21                  dfs_visit(W,bool,i);
22              }
23      }//dfs 함수 시작
24
25          for(int i=0;i<W.length;i++) {
26          System.out.printf("인덱스 번호 : %d 탐색 시작 시간 : %2d ",i,W[i].depart);
27          System.out.printf("탐색 종료 시간 : %2d ",W[i].finish);
28          System.out.println("부모노드 : "+W[i].parent);
29          }
30      }
31      public static void dfs_visit(node[] u,boolean[][] bool,int index){
32          time=time+1;
33          u[index].depart = time;
34          u[index].color = "gray";
35          for(int k =0;k<bool.length;k++) {
36              if(bool[index][k]==true)
37                  if(u[k].color.equals("white")) {//아직 미방문일경우
38                      u[k].parent=index;//부모노드를 설정하고
39                      dfs_visit(u,bool,k);//방문한다.
40                  }
41          }
42          u[index].color="black";//주변 이웃 방문이 끝나고 black으로 하고
43          time=time+1;
44          u[index].finish=time;//끝난시간을 저장한다
45      }
```

노드의 연결상태를 boolean배열로 선언하고 각 노드들의 상태를 초기화한다 첫 번째 시작노드를 0로 선택한다.

dfs_visit 함수는 시작시 time을 저장하고 들어온 노드배열,연결확인용 bool배열과 노드의 인덱스를 가지고 bool배열을 이용해 연결되었고 아직 미방문인 노드를 찾아서 부모노드를 설정하고 방문한다.

다 끝난후 time을 증가시킨다.

```
static class node {
    int     depart;
    int     parent;
    int     finish;
    String  color;
     public node(int d,int f,int x,String y){
         this.depart=d;
         this.finish=f;
         this.parent=x;
         this.color = y;
     }

}
```

노드 객체는 시작 시간과 끝시간 부모노드 컬러(방문여부)를 저장한다.

2.실행결과

3. 느낀 점

연결 여부를 bool배열로 선언후 활용하였더니 쉽게 구현되었다. DFS의 큐를 넣고 넣은 노드의 인덱스를 어떻게 저장할까 고민하다가 그냥 큐를 하나 더 만들었다. 다른방법으로도 구현이 가능할거같다.