



2023 Algorithm Study 10Week

[통계](#) [수정](#) [삭제](#)

sookyoung0620 · 방금 전 · 비공개

0

2023 Algorithm Study



목록 보기

10/10



방향그래프

- 모든 간선이 방향간선인 그래프
- 모든 간선이 한 방향으로 진행되는 그래프
- 방향 DFS 알고리즘
 - 트리간선
 - 후향간선
 - 전향간선
 - 교차간선
- 어느 정점에서든지 다른 모든 정점에 도달 가능 : 강연결
- 이행적 폐쇄
 - G^* 는 G 와 동일한 정점들로 구성
 - G 에 u 로부터 $v \neq u$ 로의 방향경로가 존재한다면 G^* 에 u 로부터 v 로의 방향 간선이 존재
 - 방향그래프에 관한 도달 가능성 정보를 제공

백준 9466

1. 순환을 이루는 4-7-6-4 확인
2. 자기 자신을 선택한 경우 확인

```

> #pragma warning (disable:4996)
#include <string.h>
#include <stdio.h>
#include <string.h>

#define MAX 100001

int N, cnt;
int want[MAX]; //팀을 원하는 학생의 번호 저장
int visited[MAX]; //방문한 노드
int done[MAX]; // 방문이 끝났는지 여부

void DFS(int nodeNum) {
    visited[nodeNum] = 1;

    int next = want[nodeNum];

    if (!visited[next])
        DFS(next);
    else if (!done[next]) {
        for (int i = next; i != nodeNum; i = want[i])
            cnt++;
        cnt++;
    }

    done[nodeNum] = 1; // 더 이상 해당 노드를 방문 안 해도 됨
}

void init() {
    for (int i = 0; i < MAX; i++) {
        visited[i] = 0;
        done[i] = 0;
    }
}

int main(void) {
    int T;
    scanf("%d", &T);

    for (int i = 0; i < T; i++) {
        init();

        scanf("%d", &N);

        for (int j = 1; j <= N; j++)
            scanf("%d", &want[j]);

        cnt = 0;

        for (int j = 1; j <= N; j++)
            if (!visited[j])
                DFS(j); // 팀 만들어진 사람 세기

        printf("%d\n", N - cnt);
    }
}

```

```
    return 0;  
}
```

-> 굳이 두 경우를 나눌 필요없다는 걸 깨달음..

-> DFS로 해결..

-> 방문된 상태지만 방문이 끝나지 않았을 때를 순환이라고 판단

-> want[MAX]; //팀을 원하는 학생의 번호 저장

visited[MAX]; //방문한 노드

done[MAX]; // 방문이 끝났는지 여부



윤수경



이전 포스트

2023 Algorithm Study 9Week

0개의 댓글

댓글을 작성하세요

댓글 작성



Powered by
Stellate