



Week10_과제 (9466)

[통계](#) [수정](#) [삭제](#)

yuyu7123 · 방금 전 · 비공개

0

baekjoon

Smarcle_23Algorithm



▼ 목록 보기

10/10



Baekjoon_9466 : 텀 프로젝트

방향그래프로 구현

풀이 과정

1. n개의 간선 (j, m)을 추가 $1 \leq j \leq n$, n번 m을 입력받음
2. 깊이 우선탐색을 이용하여 각 노드를 탐색, 시작 정점과 끝 정점이 같다면 1을 반환, 아니면 0을 반환
3. n에서 1을 반환한 노드의 개수를 빼 팀에 속하지 않은 노드의 수를 출력

문제 해결

- 시간초과
깊이 우선 탐색을 재귀적으로 호출하여 발생한 것으로 추정, 반복문으로 변경 -> 시간초과
- 메모리 해제를 추가하였지만 1%에서 시간초과 π
n이 큰 값이 들어올 때가 문제인 거 같은데 무엇을 수정해야할 지 감이 안옴

코드

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
```

```
#include <stdlib.h>
#define MAX_SIZE 100001

typedef struct Edge {
    int to;
    struct Edge* next;
} Edge;

Edge* edges[MAX_SIZE];
int visited[MAX_SIZE];

Edge* getEdge(int to) {
    Edge* ret = (Edge*)malloc(sizeof(Edge));
    ret->to = to;
    ret->next = NULL;

    return ret;
}

void addEdge(int from, int to) {
    Edge* edge = getEdge(to);
    edge->next = edges[from];
    edges[from] = edge;
}

int dfs(int start, int cur) {
    int stack[MAX_SIZE];
    int top = -1;

    visited[cur] = 1;
    stack[++top] = cur;

    while (top >= 0) {
        int current = stack[top--];
        int next = edges[current]->to;

        if (!visited[next]) {
            visited[next] = 1;
            stack[++top] = next;
        }
        else if (visited[next] && next == start) {
            return 1;
        }
    }

    return 0;
}

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

    for (int i = 0; i < T; i++) {
        scanf("%d", &n);
        for (int j = 1; j <= n; j++) {
            scanf("%d", &m);
            addEdge(j, m);
        }
    }
}
```

```
int failCount = 0;

for (int j = 1; j <= n; j++) {
    for (int k = 1; k <= n; k++) {
        visited[k] = 0;
    }
    failCount += dfs(j, j);
}

printf("%d\n", n - failCount);

for (int i = 1; i < MAX_SIZE; i++) {
    Edge* current = edges[i];
    while (current != NULL) {
        Edge* next = current->next;
        free(current);
        current = next;
    }
    edges[i] = NULL;
}

return 0;
}
```



유정



이전 포스트

Week9_과제 (5567)

0개의 댓글

댓글을 작성하세요

댓글 작성

