

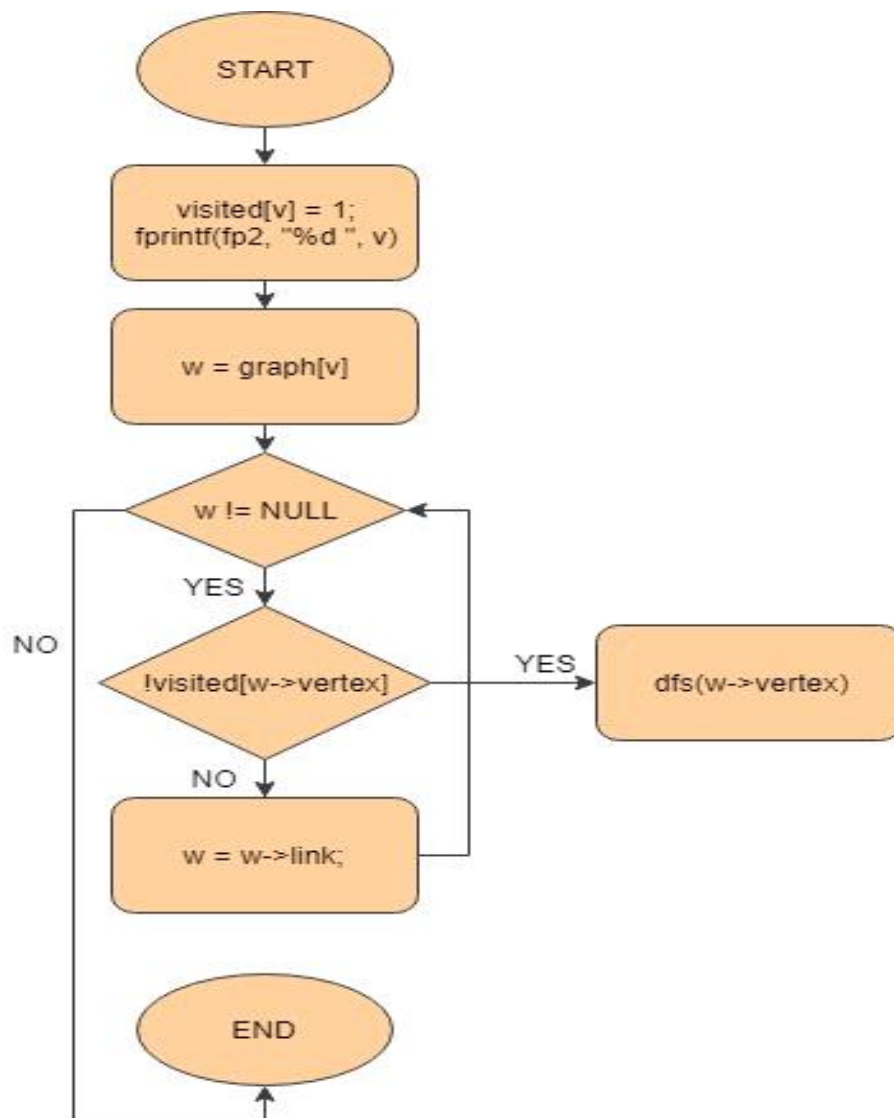
자료구조 (Data Structure) 결과 보고서

Programming Assignment 6

컴퓨터공학과 20181593 계인혜

1. Write a complete code that outputs all of the connected components of an undirected graph $G = (V, E)$. The graph does not have self-loops and multiple occurrences of the same edge. An input text file that has an adjacency matrix of a graph is given. Your program needs to read the input file and transform the adjacency matrix into the adjacency list representation and then your DFS should work on the graph represented by the adjacency list.

① 순서도



② 알고리즘 설명 및 자료구조

파일 포인터 fp1, fp2를 이용하여 각각 읽기 모드와 쓰기모드로 선언하였다. 파일이 잘 열리지 않았다면 예외 문장을 출력할 수 있도록 처리하였다. 그 다음으로 input.txt 파일에서 n을 입력받아 그 크기만큼 visited배열을 동적으로 선언한다. graph는 자료형이 node_pointer인 최대 크기가 50인 배열이다. 본격적으로 자료를 입력받기 전에 visited 배열을 0으로, 각 정점의 데이터를 자기 자신으로 초기화 시켜놓는다. 이후 $n * n$ 행렬의 값을 하나씩 입력 받아 저장하는데 그 값이 1인 경우에 $ptr = graph[i]$ 로 설정하고 node_pointer w의 vertex에 j를 저장한 후 link는 NULL로 설정한다. 이때 i, j 는 각각 input.txt에서의 행과 열을 의미한다. 반복문을 이용하여 ptr의 맨 마지막 자리에 w를 삽입한다. 이렇게 인접 행렬로 주어진 그래프를 인접 리스트 표현으로 바꾸고 나면 dfs를 실행할 수 있게 된다. 상단에 첨부한 순서도의 알고리즘을 따라 깊이 우선 탐색을 실행하여 output.txt에 결과를 출력한다. 마지막으로 동적으로 할당했던 visited 배열을 free 하고, 읽기 모드와 쓰기 모드로 열었던 파일 포인터들을 닫아준다.

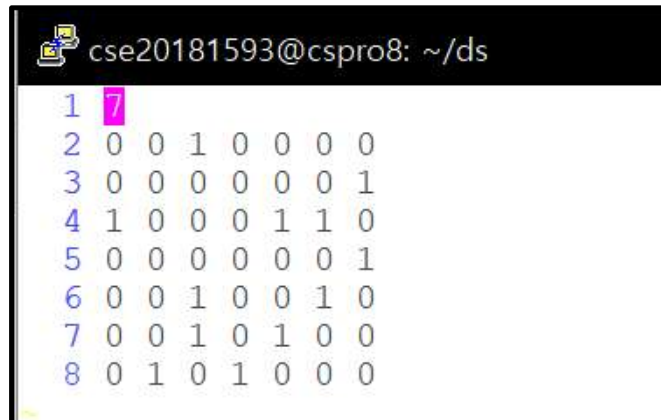
```
typedef struct node *node_pointer;
typedef struct node {
    int vertex;
    node_pointer link;
}node;

node_pointer* graph;
int* visited;
```

③ pseudo code

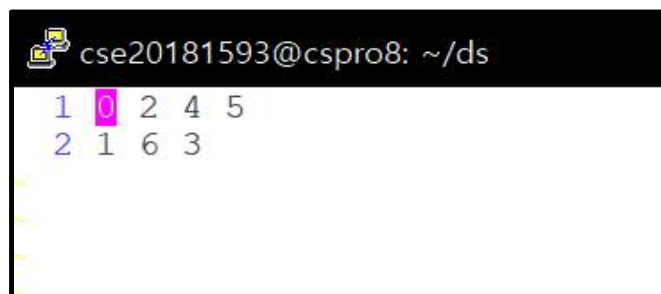
```
Open the file pointer.  
Initialize a visited array to false and graph array.  
if(num == 1) {  
    ptr = graph[i];  
    malloc the node_pointer w.  
    find a correct location to insert w.  
    insert the node_pointer w to ptr's tail.  
}  
For every vertex if(visited[vertex] == 0) call the dfs function.  
Free visited array and close the file pointer.
```

④ testcase



```
cse20181593@csp8: ~/ds  
1 7 0 0 0 0 0 0  
2 0 0 1 0 0 0 0  
3 0 0 0 0 0 0 1  
4 1 0 0 0 1 1 0  
5 0 0 0 0 0 0 1  
6 0 0 1 0 0 1 0  
7 0 0 1 0 1 0 0  
8 0 1 0 1 0 0 0
```

input.txt



```
cse20181593@csp8: ~/ds  
1 0 2 4 5  
2 1 6 3
```

output.txt