**Homework 4B report**

**2014313303 홍태하(개발환경 : Linux)**

1. **알고리즘**

Animal이라는 구조체를 만들었는데 animal의 name, rank, num, indegree를 저장한다. Name은 말그대로 이름을 저장하는 것이고 rank는 그 동물의 rank가 몇인지 저장한다. Num은 adjacency matrix에서의 index와 같은 것이고 indegree는 이 동물을 가리키고 있는 다른 동물들의 수이다.

먼저 strtok를 사용해서 인풋을 받는다. 인풋 정보를 토대로 adjacency matrix를 만든다. 그 후 DFS를 이용하여 cycle이 있는지 체크하고 cycle이 있으면 Stupid David!를 출력하고 끝난다. Cycle이 없다면 BFS를 이용하여 동물들의 rank를 정해준다. 마지막으로 동물들의 rank와 name을 기준으로 sorting을 해준 후 출력한다.

1. **코드 설명.**
2. **인풋 받기.**

먼저 전체 동물의 개수 N과 condition의 개수 C를 받는다. 그 후 Animal구조체를 저장할 배열 animal을 N개의 공간만큼 할당한다. 또 cycle 체크에 사용할 visited배열도 N만큼 할당하고 WHITE로 초기화한다. 인풋의 두번째 줄은 전체 동물의 이름이 ‘/’로 구분되어 나열되어 있다. 동물 이름의 최대 길이는 19이고 동물은 최대 99마리가 있을 수 있으므로 19\*99인데 이 경우 ‘/’의 개수는 98이다. 또 fgets를 사용하여 인풋을 받을 것이기 때문에 맨 끝에 NULL과 ‘\n’이 들어갈 자리까지 생각하면 19\*99 + 98 +2 = 1981만큼의 공간이 필요하다. 따라서 MAX\_INPUT을 1981이라고 define하고 배열 input에 동물들의 이름을 받았다. 그리고 strtok를 이용해서 동물들의 이름을 받았고 rank와 indegree는 0으로, num은 index와 같게 초기화하여 animal 배열에 저장하였다.

N\*N크기의 adj\_arr배열을 할당하고 0으로 초기화하였다. 이것은 adjacency matrix이다. 위와 같은 방식으로 condition을 fgets를 이용해 받은 후 strtok로 잘라서 animal배열의 동물 이름과 비교하였다. condition에서 왼쪽에 있는 동물의 index는 strong에 저장하고 오른쪽에 있는 동물의 index는 weak에 저장하였다. 그 후 adj\_arr배열의 strong과 weak의 위치에 맞는 곳을 1로 바꿔주고 weak인 동물의 indegree를 1증가시켰다. 이렇게 인풋을 받아서 animal 배열과 adj\_arr 배열을 완성하였다.

1. **Cycle 체크.**

Cycle 체크는 IsCycle\_DFS함수와 IsCycle\_DFS\_Visit 함수 두개로 이루어져있다. Cycle 체크는 DFS를 이용해서 했는데 여기서 사용되는 visited 배열은 color를 저장한다. Color가 WHITE이면 아직 한번도 방문하지 않은 것이고 GRAY이면 방문했지만 아직 끝나지 않은 것이다. BLACK이면 완전히 끝난 것이다. Visited의 index는 animal의 num과 일치한다.

IsCycle\_DFS함수에서는 visited가 WHITE인 index를 찾아 IsCycle\_DFS\_Visit 함수를 호출한다. 이 함수가 불리면 방문 된 것이므로 GRAY로 표시하고 방문 된 vertex는 u라고 한다. Vertex u와 연결된 다른 vertex v를 찾는다. V가 만약 WHITE라면 IsCycle\_DFS\_Visit함수를 다시 불러서 더 안쪽 vertex로 들어가게 된다. 하지만 v가 GRAY라면 (u,v) edge는 Back edge가 되며 이것은 cycle을 생성한다. GRAY vertex를 만났다는 것은 아직 끝나지 않았는데 그 곳으로 다시 돌아갔다는 것이므로 cycle을 의미한다. 이 경우 true를 리턴하며 cycle인 것을 알리고 종료한다. 재귀함수로 짜여져 있으므로 각 함수의 리턴값을 저장했다가 true이면 바로 true를 리턴해주어야 한다. 만약 WHITE와 GRAY 둘 다 만나지 않았다면 끝난 것이므로 BLACK으로 바꿔주고 cycle이 없는 것이므로 false를 리턴하게 된다.

cycle임을 알았다면 Stupid David!를 출력하고 동적할당 된 배열 animal과 adj\_arr를 free해준 후 종료한다.

1. **Rank 정하기.**

Rank는 BFS를 이용하여 정하였다. 먼저 animal배열에서 indegree가 0인 것들을 찾았다. 이것은 이 동물들에게는 상위 ranker가 없다는 뜻이므로 이들의 rank는 전부 1이 된다. BFS를 사용하기 위해 Queue를 이용하였다. Rank가 1인 동물들에서 시작하여 한 칸 씩 내려가며 rank를 정해주고 그 중에 큰 것을 고르도록 하였다. 코드를 자세히 보면 먼저 rank가 1인 동물을 queue에 넣는다. current변수는 현재 같은 distance에 있는 동물들의 수이고 next변수는 다음 distance에 있는 동물들의 수이다. Queue가 empty가 될 때까지 while문을 반복하며 queue에서 하나 뽑아 u에 저장한다. 현재 distance에서 하나가 뽑힌 것이므로 current는 1 감소한다. U와 연결되어 있는 vertex v를 찾아 queue에 넣는다. 이들은 다음 distance의 vertex이므로 next를 1씩 증가시켜준다. Temp\_rank는 해당 rank가 1인 노드에서 시작했을 때 distance가 증가함에 따라 1씩 증가한 값이다. 하지만 다른 rank가 1인 노드에서 시작했을 때 더 높은 rank값을 받았을 수 있으므로 temp\_rank와 현재 rank의 값을 비교하여 큰 값을 취한다. 이 과정을 반복하여 current가 0이 되면 현재 distance의 모든 vertex를 처리한 것이므로 current를 next로 바꿔주고 next는 0으로 temp\_rank는 1증가시킴으로써 다음 distance로 넘어가 같은 작업을 반복한다. 이것을 모든 rank가 1인 vertex에서 하면 동물들의 rank가 정해진다.

1. **Sort.**

Sort는 stdlib.h에서 지원하는 qsort를 이용해서 하였다. 먼저 rank를 기준으로 rank값이 작은 것이 앞에 오게 하였고 rank가 같은 경우 strcmp로 동물들의 name을 비교하여 사전순으로 나열되게 하였다.

1. **예시**

인풋이 이렇게 들어온 경우 Tiger는 animal[0]에 Lion은 animal[1]에 저장되고 adj\_arr는 가 된다. Cycle 체크를 하면 먼저 Tiger가 WHITE이므로 IsCycle\_DFS\_Visit함수에 들어간다. Tiger가 GRAY로 바뀌고 Tiger와 연결된 vertex를 찾는다. Lion뿐이므로 Lion이 v가 되고 Lion은 WHITE이므로 다시 IsCycle\_DFS\_Visit함수가 불린다. Lion도 GRAY로 바뀌고 연결된 다른 vertex를 찾지만 없으므로 Lion은 BLACK이 되고 false를 리턴한다. Tiger도 BLACK으로 바뀌고 false를 리턴한다. IsCycle\_DFS로 돌아와서 더 이상 WHITE인 vertex는 없고 리턴값도 true가 아니었으므로 false를 리턴하며 종료된다. 이것은 cycle이 없는 것이므로 rank를 정한다. indegree가 0인 Tiger에서부터 시작한다. Tiger를 queue에 넣고 current를 1, next를 0으로 한 후 queue가 empty일때까지 반복한다. queue에서 Tiger를 빼고 current를 0으로 한 후 Tiger와 연결된 vertex를 찾는다. Lion뿐이므로 queue에 넣고 rank를 2로 바꿔준다. Next를 1로 해준 후 current가 0이 됐으므로 current를 next로, next를 0으로, temp\_rank를 3으로 바꿔준 후 다음 distance로 넘어간다. Lion이 뽑히고 current는 0이 되고 Lion과 연결된 vertex를 찾는다. 하지만 Lion과 연결된 다른 vertex가 없으므로 이대로 종료되고 indegree가 0인 동물도 Tiger뿐이므로 rank정하기가 끝난다. Sorting해준 후 결과를 출력하면 이렇게 된다.

1 Tiger

2 Lion

2 1

Tiger/Lion

Tiger/Lion