

그래프 1 (도전)

최백준 choi@startlink.io

BFS 스페셜 저지

<https://www.acmicpc.net/problem/16940>

- 입력으로 주어진 정보를 이용해서 BFS 탐색을 진행하면서 올바른 결과인지 살펴본다.

BFS 스페셜 저지

<https://www.acmicpc.net/problem/16940>

```
int m = 1; // 큐의 크기
for (int i=0; i<n; i++) {
    if (q.empty()) // 아직 BFS가 진행 중인데 큐가 비어있음
        int x = q.front(); q.pop();
    if (x != order[i]) // 순서가 올바르지 않음
        int cnt = 0; // 이번에 큐에 넣어야 할 정점의 수
    for (int y : a[x]) { if (check[y] == false) { parent[y] = x; cnt +=
1; } }
    for (int j=0; j<cnt; j++) {
        if (m+j >= n || parent[order[m+j]] != x) // x와 연결되지 않은 정점이
큐에 들어가있으니 올바르지 않음
            q.push(order[m+j]); check[order[m+j]] = true;
    }
    m += cnt;
}
```

BFS 스페셜 저지

<https://www.acmicpc.net/problem/16940>

- 소스: <http://codeplus.codes/6c58c942cff74fc39d27f890916cce95>

BFS 스페셜 저지

<https://www.acmicpc.net/problem/16940>

- 입력으로 주어진 순서를 이용해서 인접 리스트를 정렬하는 방식도 가능하다.

```
for (int i=0; i<n; i++) {  
    cin >> b[i];  
    b[i] -= 1;  
    order[b[i]] = i;  
}  
  
for (int i=0; i<n; i++) {  
    sort(a[i].begin(), a[i].end(), [&](const int &u, const int &v) {  
        return order[u] < order[v];  
    });  
}
```

BFS 스페셜 저지

<https://www.acmicpc.net/problem/16940>

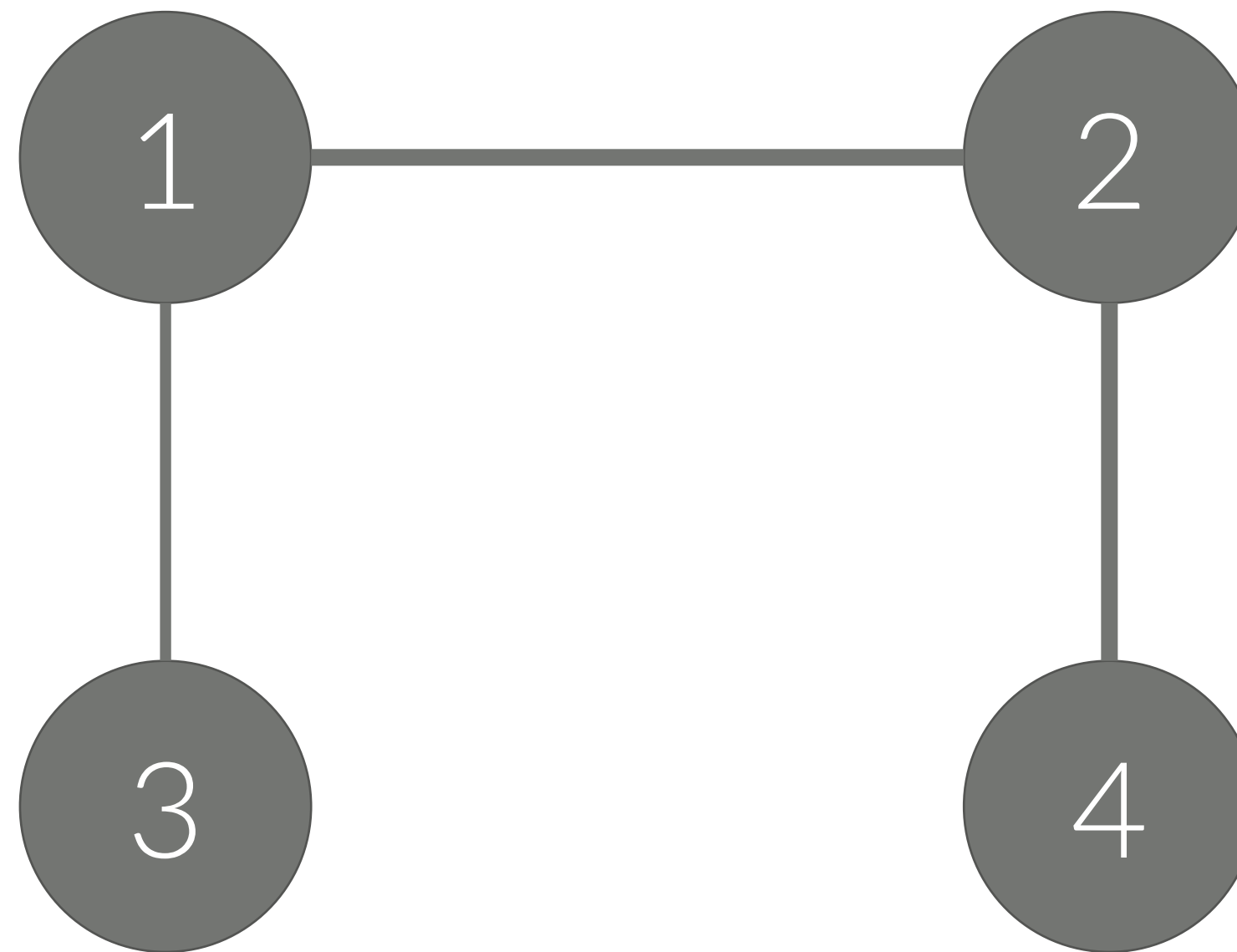
- 소스: <http://codeplus.codes/9107adffea9a4008810ad7cd19d59301>

DFS 스페셜 저지

7

<https://www.acmicpc.net/problem/16964>

- 트리와 1에서 시작한 DFS의 결과가 주어졌을 때, 이 탐색 결과가 올바른 것인지 알아보는 문제
- 올바른 순서
- 1, 2, 4, 3
- 1, 3, 2, 4



DFS 스페셜 저지

<https://www.acmicpc.net/problem/16964>

- 입력으로 주어진 순서를 이용해서 인접 리스트를 정렬한 다음, 이 순서로 DFS 탐색을 진행해본다.

```
for (int i=0; i<n; i++) {  
    cin >> b[i];  
    b[i] -= 1;  
    order[b[i]] = i;  
}  
  
for (int i=0; i<n; i++) {  
    sort(a[i].begin(), a[i].end(), [&](const int &u, const int &v) {  
        return order[u] < order[v];  
    });  
}
```


DFS 스페셜 저지

<https://www.acmicpc.net/problem/16964>

- 소스: <http://codeplus.codes/65c9350c2ff44ffc8e288b1d502533d9>

다리 만들기

<https://www.acmicpc.net/problem/2146>

- 여러 섬으로 이루어진 나라에서
- 두 섬을 연결하는 가장 짧은 다리를 찾는 문제

다리 만들기

<https://www.acmicpc.net/problem/2146>

- 단지번호붙이기 + 토마토 문제
- 먼저, 섬을 그룹을 나눈다
- $g[i][j] = (i,j)$ 의 그룹 번호
- 그 다음 각각의 그룹에 대해서 다른 섬까지 거리를 계산한다
- 이 방법은 각각이 그룹에 대해서 BFS 알고리즘을 수행해야 하기 때문에 느리다

다리 만들기

12

<https://www.acmicpc.net/problem/2146>

- 소스: <http://codeplus.codes/7dcd9a83669e41cab4b00088a0c679e0>

- 더 빠른 알고리즘으로 땅을 확장하는 방식을 생각해 볼 수 있다.

[illegible]

다리 만들기

<https://www.acmicpc.net/problem/2146>

- 왼쪽: 그룹, 오른쪽: 거리

1	1	1					2	2	2
1	1	1	1					2	2
1		1	1					2	2
		1	1	1					2
			1						2
									2
				3	3				
				3	3	3			

0	0	0	-	-	-	-	0	0	0
0	0	0	0	-	-	-	-	0	0
0	-	0	0	-	-	-	-	0	0
-	-	0	0	0	-	-	-	-	0
-	-	-	0	-	-	-	-	-	0
-	-	-	-	-	-	-	-	-	0
-	-	-	-	-	-	-	-	-	-
-	-	-	-	0	0	-	-	-	-
-	-	-	-	0	0	0	-	-	-
-	-	-	-	-	-	-	-	-	-

다리 만들기

15

<https://www.acmicpc.net/problem/2146>

- 왼쪽: 그룹, 오른쪽: 거리

1	1	1	1			2	2	2	2
1	1	1	1	1			2	2	2
1	1	1	1	1			2	2	2
1	1	1	1	1	1			2	2
		1	1	1				2	2
			1					2	2
				3	3				2
			3	3	3	3			
			3	3	3	3	3		
				3	3	3			

0	0	0	1	-	-	1	0	0	0
0	0	0	0	1	-	-	1	0	0
0	1	0	0	1	-	-	1	0	0
1	1	0	0	0	1	-	-	1	0
-	-	1	0	1	-	-	-	1	0
-	-	-	1	-	-	-	-	1	0
-	-	-	-	1	1	-	-	-	1
-	-	-	1	0	0	1	-	-	-
-	-	-	1	0	0	0	1	-	-
-	-	-	-	1	1	1	-	-	-

다리 만들기

<https://www.acmicpc.net/problem/2146>

- 왼쪽: 그룹, 오른쪽: 거리

1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	1	2	2	2	2
1	1	1	1	1	1	2	2	2	2
1	1	1	1	1	1	1	2	2	2
1	1	1	1	1	1		2	2	2
		1	1	1	3		2	2	2
			1	3	3	3		2	2
		3	3	3	3	3	3		2
		3	3	3	3	3	3	3	
			3	3	3	3	3		

0	0	0	1	2	2	1	0	0	0
0	0	0	0	1	2	2	1	0	0
0	1	0	0	1	2	2	1	0	0
1	1	0	0	0	1	2	2	1	0
2	2	1	0	1	2	-	2	1	0
-	-	2	1	2	2	-	2	1	0
-	-	-	2	1	1	2	-	2	1
-	-	2	1	0	0	1	2	-	2
-	-	2	1	0	0	0	1	2	-
-	-	-	2	1	1	1	2	-	-

다리 만들기

<https://www.acmicpc.net/problem/2146>

- 왼쪽: 그룹, 오른쪽: 거리

1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	1	2	2	2	2
1	1	1	1	1	1	2	2	2	2
1	1	1	1	1	1	1	2	2	2
1	1	1	1	1	1	1	2	2	2
1	1	1	1	1	3	2	2	2	2
		1	1	3	3	3	2	2	2
	3	3	3	3	3	3	3	2	2
	3	3	3	3	3	3	3	3	2
		3	3	3	3	3	3	3	

0	0	0	1	2	2	1	0	0	0
0	0	0	0	1	2	2	1	0	0
0	1	0	0	1	2	2	1	0	0
1	1	0	0	0	1	2	2	1	0
2	2	1	0	1	2	3	2	1	0
3	3	2	1	2	2	3	2	1	0
-	-	3	2	1	1	2	3	2	1
-	3	2	1	0	0	1	2	3	2
-	3	2	1	0	0	0	1	2	3
-	-	3	2	1	1	1	2	3	-

다리 만들기

<https://www.acmicpc.net/problem/2146>

- 왼쪽: 그룹, 오른쪽: 거리

1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	1	2	2	2	2
1	1	1	1	1	1	2	2	2	2
1	1	1	1	1	1	1	2	2	2
1	1	1	1	1	1	1	2	2	2
1	1	1	1	1	3	2	2	2	2
1	1	1	1	3	3	3	2	2	2
3	3	3	3	3	3	3	3	2	2
3	3	3	3	3	3	3	3	3	2
	3	3	3	3	3	3	3	3	2

0	0	0	1	2	2	1	0	0	0
0	0	0	0	1	2	2	1	0	0
0	1	0	0	1	2	2	1	0	0
1	1	0	0	0	1	2	2	1	0
2	2	1	0	1	2	3	2	1	0
3	3	2	1	2	2	3	2	1	0
4	4	3	2	1	1	2	3	2	1
4	3	2	1	0	0	1	2	3	2
4	3	2	1	0	0	0	1	2	3
-	4	3	2	1	1	1	2	3	4

다리 만들기

<https://www.acmicpc.net/problem/2146>

- 왼쪽: 그룹, 오른쪽: 거리

1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	1	2	2	2	2
1	1	1	1	1	1	2	2	2	2
1	1	1	1	1	1	1	2	2	2
1	1	1	1	1	1	1	2	2	2
1	1	1	1	1	3	2	2	2	2
1	1	1	1	3	3	3	2	2	2
3	3	3	3	3	3	3	3	2	2
3	3	3	3	3	3	3	3	3	2
3	3	3	3	3	3	3	3	3	2

0	0	0	1	2	2	1	0	0	0
0	0	0	0	1	2	2	1	0	0
0	1	0	0	1	2	2	1	0	0
1	1	0	0	0	1	2	2	1	0
2	2	1	0	1	2	3	2	1	0
3	3	2	1	2	2	3	2	1	0
4	4	3	2	1	1	2	3	2	1
4	3	2	1	0	0	1	2	3	2
4	3	2	1	0	0	0	1	2	3
5	4	3	2	1	1	1	2	3	4

다리 만들기

<https://www.acmicpc.net/problem/2146>

- 각 칸과 인접한 칸의 그룹 번호가 다르면 다리를 만들 수 있다

1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	1	2	2	2	2
1	1	1	1	1	1	2	2	2	2
1	1	1	1	1	1	1	2	2	2
1	1	1	1	1	1	1	2	2	2
1	1	1	1	1	3	2	2	2	2
1	1	1	1	3	3	3	2	2	2
3	3	3	3	3	3	3	3	2	2
3	3	3	3	3	3	3	3	3	2
3	3	3	3	3	3	3	3	3	2

0	0	0	1	2	2	1	0	0	0
0	0	0	0	1	2	2	1	0	0
0	1	0	0	1	2	2	1	0	0
1	1	0	0	0	1	2	2	1	0
2	2	1	0	1	2	3	2	1	0
3	3	2	1	2	2	3	2	1	0
4	4	3	2	1	1	2	3	2	1
4	3	2	1	0	0	1	2	3	2
4	3	2	1	0	0	0	1	2	3
5	4	3	2	1	1	1	2	3	4

다리 만들기

<https://www.acmicpc.net/problem/2146>

- 길이: 4

1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	1	2	2	2	2
1	1	1	1	1	1	2	2	2	2
1	1	1	1	1	1	1	2	2	2
1	1	1	1	1	1	1	2	2	2
1	1	1	1	1	3	2	2	2	2
1	1	1	1	3	3	3	2	2	2
3	3	3	3	3	3	3	3	2	2
3	3	3	3	3	3	3	3	3	2
3	3	3	3	3	3	3	3	3	2

0	0	0	1	2	2	1	0	0	0
0	0	0	0	1	2	2	1	0	0
0	1	0	0	1	2	2	1	0	0
1	1	0	0	0	1	2	2	1	0
2	2	1	0	1	2	3	2	1	0
3	3	2	1	2	2	3	2	1	0
4	4	3	2	1	1	2	3	2	1
4	3	2	1	0	0	1	2	3	2
4	3	2	1	0	0	0	1	2	3
5	4	3	2	1	1	1	2	3	4

다리 만들기

<https://www.acmicpc.net/problem/2146>

- 길이: $2+1 = 3$

1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	1	2	2	2	2
1	1	1	1	1	1	2	2	2	2
1	1	1	1	1	1	1	2	2	2
1	1	1	1	1	1	1	2	2	2
1	1	1	1	1	3	2	2	2	2
1	1	1	1	3	3	3	2	2	2
3	3	3	3	3	3	3	3	2	2
3	3	3	3	3	3	3	3	3	2
3	3	3	3	3	3	3	3	3	2

0	0	0	1	2	2	1	0	0	0
0	0	0	0	1	2	2	1	0	0
0	1	0	0	1	2	2	1	0	0
1	1	0	0	0	1	2	2	1	0
2	2	1	0	1	2	3	2	1	0
3	3	2	1	2	2	3	2	1	0
4	4	3	2	1	1	2	3	2	1
4	3	2	1	0	0	1	2	3	2
4	3	2	1	0	0	0	1	2	3
5	4	3	2	1	1	1	2	3	4

다리 만들기

<https://www.acmicpc.net/problem/2146>

- 소스: <http://codeplus.codes/dc3ddf21fff74a649925847fdb0afe4>

끝

코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 codeplus@startlink.io 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.