

void init(**int** N, **int** M, **int** roads[][2])

N: 마을 개수(≤ 50) , M : 도로 개수(≤ 50)
roads : 양방향 도로

마을 개수 : 50
카페 개수 : 10,000

void addCafe(**int** townID, **char** name[]) : 10,000회

townID(1~N) 마을에 name(소문자 3~5) 카페 추가

void addScore(**char** name[], **int** score) : 10,000회

name 가치 += score

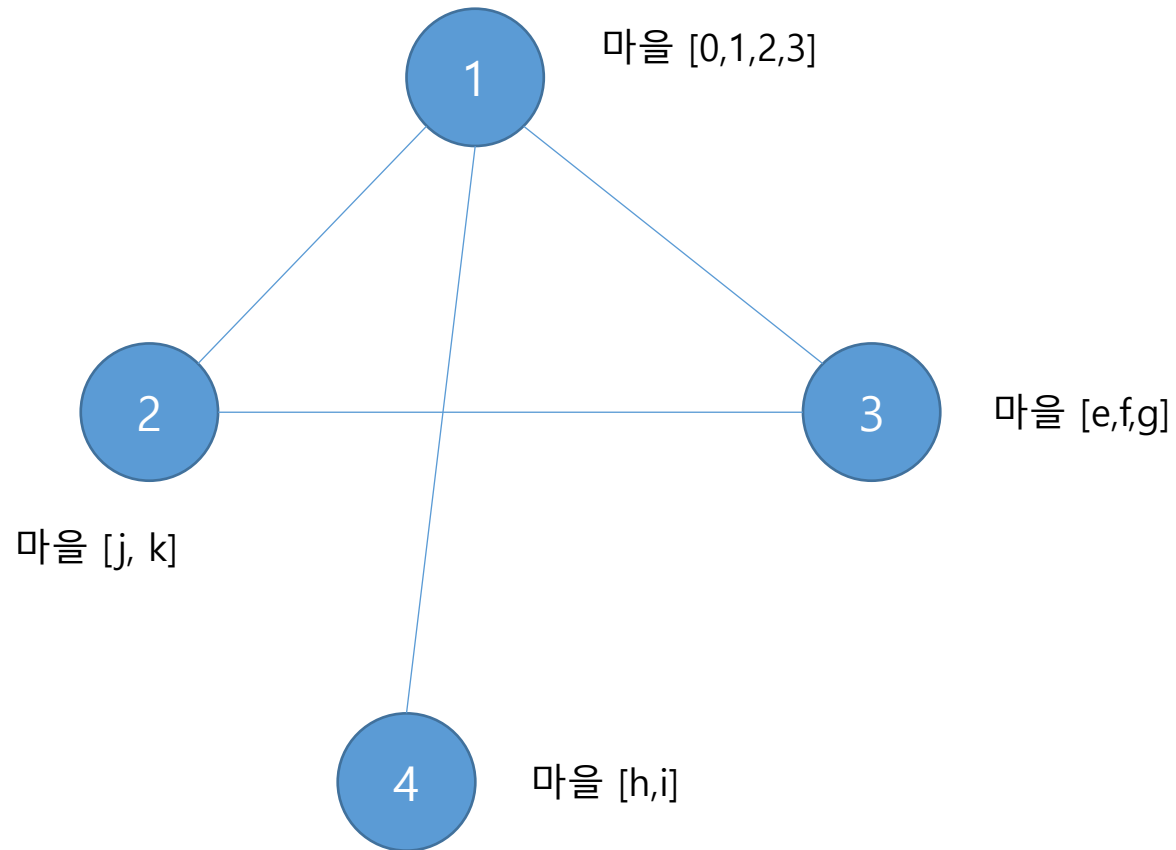
int getBestScore(**char** str[]) : 150,000회

이름에 str(소문자 1~5)이 포함되는 카페들 중 최고 가치 구하기

int getTop3Score(**int** townID, **int** step) : 1,000회

townID 마을과 step 거리 이내의 마을들의 모든 카페 중
가장 높은 가치 3개의 합 구하기

1. 후보군 줄이기
2. 미리 저장하는 방법



a

카페	score	town
a	1	1
b	10	1
c	55	1
d	20	1
e	30	2
f	60	2
g	7	2
..		

그래프

인접 행렬 : list adj[53][53]

인접 리스트 : list adj[53] []

마을의 카페 리스트 : list cafe[53]

<code>void addScore(char name[], int score)</code>	: 10,000회
<code>int getBestScore(char str[])</code>	: 150,000회

`str` = 소문자 1~5

`str`을 포함하는 카페의 최대 `score`

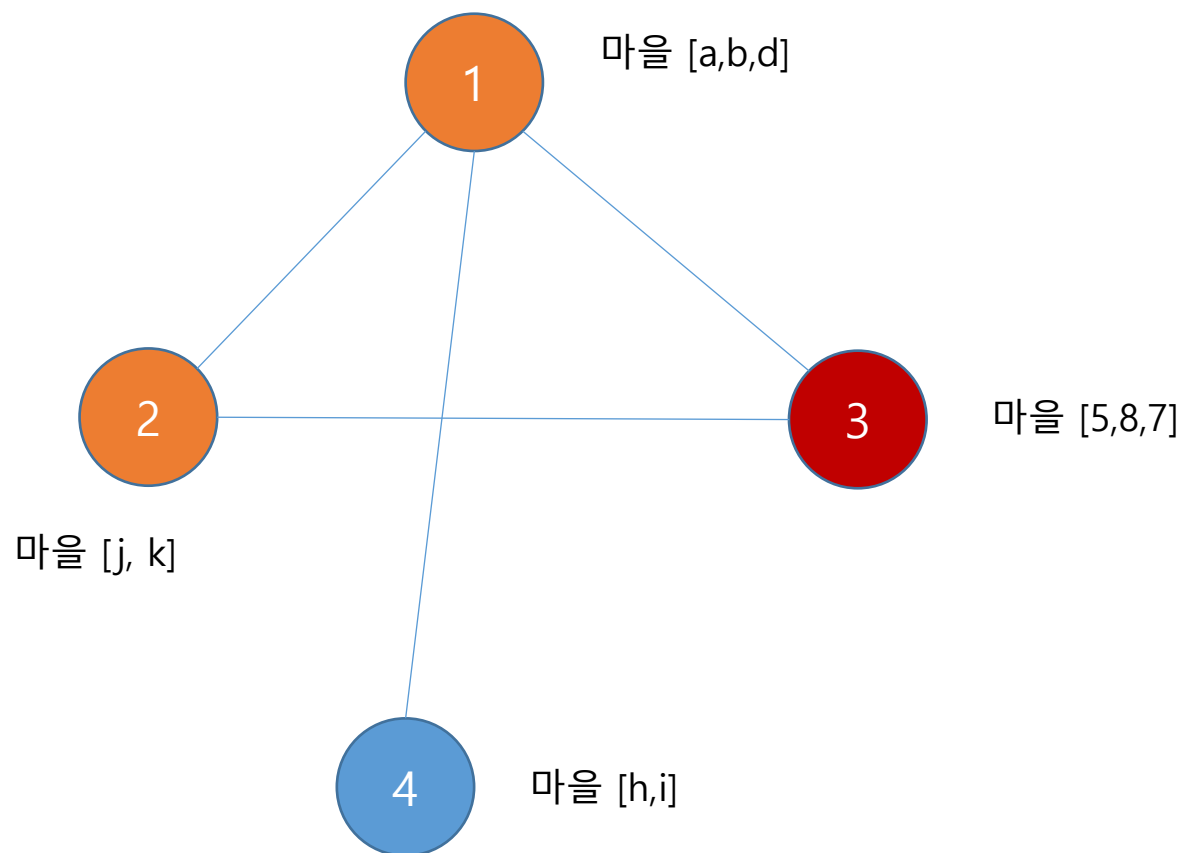
`name("abcde") => substr 최대 15개`

a,b,c,d,e
ab,bc,cd,de
abc,bcd,cde
abcd,bcde
abcde

`maxScore[str]` = `str`을 포함하는 카페의 최대 가치
`maxScore[str] = max(maxScore[str], newScore)`

1. `name` 의 부분문자열 만들기
2. `score[name]` , `maxScore[str]`

`int` getTop3Score(`int` townID, `int` step) : 1,000회



1. step 이내의 마을 구하기 : bfs
2. 고른 마을의 카페를 후보로 하여 score 높은 3개 카페 구하기

마을 수 : 50
도로 수 : 50 (양방향=100)
카페 이름 : 5자리
str : 5자리

addCafe() : 10,000 => 카페 수 10,000개
addScore() : 10,000
dict cafeScore[name]+=score
getBestScore(str) : 150,000 , 카페 name에 str을 포함하는 카페 최대 가치
getTop3Score() : 1,000 , 특정 마을 특정 거리 이내의 카페 최대 가치

1. 마을 구성

인접한 마을
카페 정보(이름, 가치)

2. getBestScore() : $O(\text{호출횟수} * \text{카페 수} * \text{len})$ => time limit
cafe name: abcde, abcd, abde, bcde
str: bc
maxScore[str] : str을 부분분자열로 하는 카페들 중 최대 가치

addScore()
score = cafeScore[name] + score
abcde
maxScore[subStr] = max(maxScore[subStr], score)
a,b,c,d,e
ab,bc,cd,de
abc,bcd,cde
abcd,bcde
abcde

3. getTop3Score()

- $O(\text{호출횟수} * \text{카페 수})$ => 괜찮을듯
- 거리 이내의 마을 선택 : BFS / 인접리스트 $O(\text{마을수} + \text{도로수})$
list adj[N+1][]
인접행렬 $O(\text{마을수} * \text{마을수})$
- 모든 마을의 모든 카페 중 최대값 3개 선택 : nlargest(), minheap 3개 유지

마을 인접리스트 : adj[townID][] : list
카페 점수 : cafeScore[name] = score : dict
마을 카페리스트 : cafeName[townID] = [cafeName, ...,] : list
str을 가지는 카페의 최대 가치 : maxScore[sub] = score : dict