

목표: 최소한 몇 번의 충전을 해야  
종점에 도착할 수 있는가?

min 충전횟수

문제 이해:

1. 한번 충전으로 이동할 수 있는 정류장의  
수가 정해져 있다.

2. 정류장 0번에서 출발하여 N번 정류장까지 이동.

3. 종점에 도착할 수 없으면 0 출력.

input

1. T: 노선 수

2.

1) K: 한번 충전으로 최대한 이동할  
수 있는 정류장 수

2) N: 종점 정류장 번호

3) M: 충전기가 설치된 정류장의 수

3. M개의 정류장 번호.

[예시]

정류장	0	1	2	3	4	5	6	7	8	9	10
충전기		0		0		0		0		0	
충전 횟수				1		2		3			

1) 완전탐색 적용 시.

매 충전소마다 충전할지 말지 정하고  
갈 수 있는지 판단.

2) 그리디 적용 시

최대한 멀리 갈 수 있는 방안으로 이동.

공통적용 사항.

첫 정류장에서 곧바로 목적지에 갈 수 있는가?

# 1) 완전 탐색 DFS 구현

① Find Value( ) :  
list\_loc\_charging ,

input 1) list: 10x: 충전기 위치, (True, False 2)

2) C: 충전 횟수 (초기값 100, min\_charging.

충전기 최대 100개)

101이면 return 0.

3) K

4) N

② Find Value 내부에 DFS 함수를  
충전 횟수 구하는 함수 구현.

DFS ( list\_loc\_charging, K, loc\_cur, ) :  
cnt: 충전 횟수,  
→ 초기값

if loc\_cur - 1 <= K :

non local min\_charging

if min\_charging > cnt :

min\_charging = cnt .

return .

else :

for i in range(1, K+1) :

if list\_loc\_charging [ loc\_cur + i ] == True :

DFS ( list\_loc\_charging - K, 1, loc\_cur + i, cnt + 1 )

## 2) 그리드로 접근

loc - cur : 현위치

i - charging : 충전소 index. 1번째 충전소

현위치에서 최대 갈수 있는 장소 쉬는 날

①  $[loc - cur - i - charging]$ :

하나라도 갈 수 없다면

②  $loc - cur + k$  까지 1씩씩 접근.  
이런 방식

①로 구현

i - charging 늘려다가 불가능하면

i - charging - 1 로 설정하고 재탐색

만약 i - charging 이 마지막이고, 이동 불가능하면  
종료