**알고리즘 스킬**

**1. 적은 한정된 범위의 인덱스를 다룰 때는 미리 배열을 만들어 보라**

***이코테 315p 그리디 기출 5번 볼링공 고르기***

Ex) 리스트에서 1~10 까지 등장하는 개수 세야 할 때

* 미리 arr = [0] \* 11 한 뒤 숫자 별로 그냥 나오는대로 1씩 증가

**2. 반복되는 심볼로 비교문 수행(경로탐색)할 때는 리스트에 저장하는 방식을 고려하자**

***이코테 110p 구현 예제 4-1 상하좌우***

Ex) UDLR 문자 입력 시 이동방향 결정하는 경우 [‘U’, ‘D’, ‘L’, ‘R’]로 저장 후 꺼내 쓰기

**3. 시간제한/메모리 제한을 고려해 데이터 수가 충분히 적다면 완전탐색을 고려해 보자**

***이코테 113p 구현 예제 4-2 시각***

* 00:00:00 ~ N시:59:59 까지 3이 포함된 시각의 개수를 셀 때 총 시각의 수는 24\*60\*60 = 10만개도 되지 않으므로 선형으로 하나씩 비교해도 1초 안에 해결이 가능하다.

**제한시간 1초 기준**

N의 범위가 500인 경우: O(N^3) 으로 해결 가능

N의 범위가 2,000인 경우 O(N^2)으로 해결 가능

N의 범위가 100,000인 경우 O(NlogN)으로 해결 가능

N의 범위가 10,000,000인 경우 O(N)으로 해결 가능

**메모리 제한**

메모리 제한 128~512MB 기준 리스트 크기 1,000만 단위 이상이면 잘못. 100만도 드뭄

**4. 숫자 포함 여부를 비교해야 하는 경우 문자열 변환을 고려해 보라.**

***이코테 113p 구현 예제 4-2 시각***

Ex) 03 25 49 (시간) 에서 3이 포함되는 경우를 찾는 경우 str(h) + str(m) + str(s) 해서

‘3’ in ‘032549’ 으로 비교

**5. 알파벳 순서를 이용하는 경우는 아스키 코드를 사용하라**

***이코테 115p 실전 문제 1 왕실의 나이트***

Ex) 소문자 알파벳을 인덱싱 하고 a를 1로 간주하려는 경우

ord(x) – ord(‘a’) + 1

**6. 이동경로를 따지는 경우 방향 하나하나 정보를 리스트에 넣자**

***이코테 115p 실전 문제 1 왕실의 나이트***

Ex) 나이트의 8자 움직임을 저장하는 경우

* [(-2,-1), (-1,-2), (1,-2) ,(2, -1), (2,1), (1,2), (-1,2), (-2,1)]

**7. 문자열 숫자, 알파벳, 대/소문자 확인 시에는 문자열 내장함수를 이용하자**

[***https://codedragon.tistory.com/9884***](https://codedragon.tistory.com/9884)

- isalpha() – 알파벳인지 확인

- isdigit(), isdecimal() – 숫자인지 확인

- isalnum() – 알파벳 또는 숫자인지 확인

- index() – 문자 위치 알려주기

- count() – 문자

- “”join(리스트) – 문자열 합치기 – “”안에 들어가는게 구분자.

**8.**