

파이썬 코딩테스트 문법 총정리

- 자주 출제되는 유형

- 그리디
- 구현
- DFS/BFS를 활용한 탐색

- 요구사항에 따른 적절한 알고리즘 설계

- N의 범위 500 : N^3
- N의 범위 2000 : N^2
- N의 범위 100000 : $N \log N$
- N의 범위 10000000 : N

- 자료형

- 실수형
round(123.456,3) # 소수점 넷째자리에서 반올림
- 기본 자료형 사칙연산
/: 나누기 연산자로 실수형으로 반환한다.
- 리스트
a = [0] * 10 # [0,0,0,0,0,0,0,0,0,0]
 - 리스트 컴프리헨션 : 초기화하는 경우에 많이 사용된다.

```
array = [i for i in range(10)]  
array = [i for i in range(10) if i%2 == 1]  
array = [i*i for i in range(10)]  
# 가장 많이 사용되는 때는 2차원 배열 초기화할 때  
array = [[0]*m for _ in range(n)]  
# m이 열이 되고 n이 행이 된다.
```

- 리스트 관련 메소드
 - append/pop : $O(1)$
 - sort()/sort(reverse=True) : $O(N \log N)$
 - 나머지는 N
- 튜플
 - 초기화는 a=(1,2,3,4)
 - 추출은 a[3] # 대괄호 사용
 - 튜플은 변경이 불가하다 !!!!!!!
- 딕셔너리
 - 데이터의 조회 및 수정에 있어서 $O(1)$ 의 시간에 처리 가능
- 집합
 - 데이터의 조회 및 수정에 있어서 $O(1)$ 의 시간에 처리 가능
- 리스트나 튜플은 순서가 있기 때문에 인덱싱을 통해 자료형의 값을 얻을 수 있다
- 딕셔너리와 집합 자료형은 key와 집합의 원소를 이용해서 1의 시간 복잡도로 조회가능

- **입력**

- `list(map(int, input.split()))`
- `a,b,c = map(int,input.split())`

- **출력**

- `print("",end= " ")` #end를 사용해서 한 줄에 출력이 가능하다
- `print(f'정답은 {answer}입니다')`

- **조건문**

- 파이썬은 $0 < x < 20$ 가능하다
- `continue` : 반복문에서 남은 코드의 실행은 건너뛰고 다음 반복을 진행
- `break` : 반복문을 즉시 탈출

- **함수**

- 한번에 여러개의 반환값이 가능하다.

- **람다**

- 특정 기능을 수행하는 함수를 한 줄에 작성이 가능하다
- `sorted(array,key=lambda x:x[1])`
- 여러 개의 리스트에 적용 가능
 - `result = map(lambda a,b:a+b,list1,list2)`