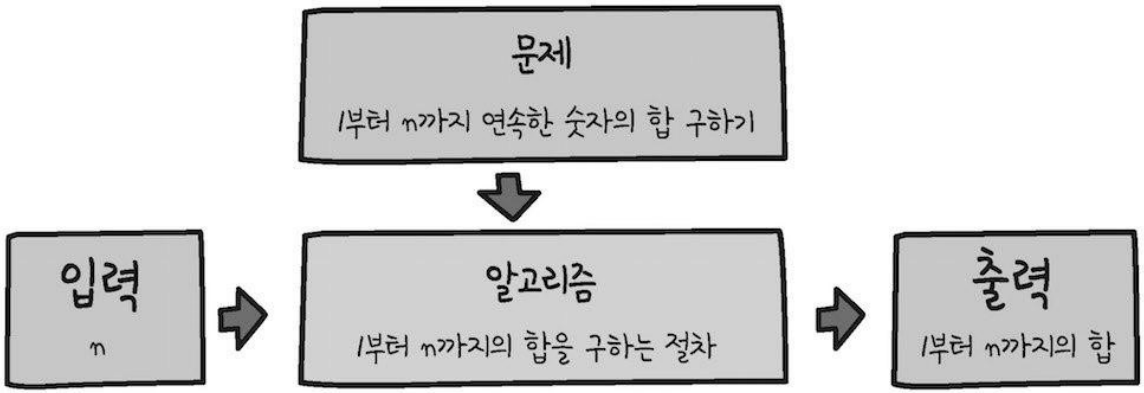
모두의 알고리즘 with 파이썬

문제 7. 순차 탐색

7 순차 탐색



[17,92,18,33,58,5,33]

특정 값의 인덱스

7 순차 탐색

알고리즘

In [1]:
```
def search_list(a, x):
    for i in range(len(a)):
        if x == a[i]:
            return i
    return -1
```

In [2]:
```
v = [17,92,18,33,58,7,33,42]
print(search_list(v, 18))
print(search_list(v, 33))
print(search_list(v, 900))
```

```
2
3
-1
```

1. 리스트의 길이 만큼 반복
2. 특정 값과 리스트 안의 숫자와 비교

1 1부터 n까지의 합 구하기

■ 7-1 리스트에서 특정 숫자의 위치를 전부 찾기

7-1

In [8]:
```
def search_all(a, x):
    return [i for i in range(len(a)) if x == a[i]]
```

In [10]:
```
print(search_all(v, 18))
print(search_all(v, 33))
print(search_all(v, 900))
```

```
[2]
[3, 6]
[]
```

특정 값과 같은 값인 인덱스(i)의 리스트 출력

1 1부터 n까지의 합 구하기

■ 7-2 프로그램 7-1의 계산 복잡도

O(n) 특정 값이 중복 될 경우를 위해, 리스트의 길이만큼 탐색

1 1부터 n까지의 합 구하기

■ 7-3 학생 번호에 해당하는 학생 이름 찾기

7-3

In [1]:
```
stu_no = [39,14,67,105]
stu_name = ["Justin","John","Mike","Summer"]
```

In [4]:
```
def name_dict(stu_no,stu_name):
    _dict = {}
    for i in range(len(stu_no)):
        _dict[stu_no[i]] = stu_name[i]
    return _dict

def get_name_from_no(name_dict,num):
    if num in name_dict.keys():
        return name_dict[num]
    else:
        return "?"
```

In [5]:
```
print(get_name_from_no(name_dict(stu_no,stu_name),39))
print(get_name_from_no(name_dict(stu_no,stu_name),14))
print(get_name_from_no(name_dict(stu_no,stu_name),67))
print(get_name_from_no(name_dict(stu_no,stu_name),333))
```

```
Justin
John
Mike
?
```

1. name_dict : 번호와 이름 쌍을 이루는 Dict 생성

2. get_name_from_on : 특정 num이 앞에 만든 Dict의 key에 존재하면 return value

3. 없을 경우 return ?

```python
def solution(participant, completion):
    participant = sorted(participant)
    completion = sorted(completion)
    completion.append("me")
    for p,c in zip(participant,completion):
        if p != c:
            return p
```

1. 참가자, 완주자 정렬
2. 완주자 1명 부족 → 마지막에 한 명 추가
3. 같은 인덱스의 값 가져와서 비교
   다를 경우 return 이름

```python
import collections


def solution(participant, completion):
    answer = collections.Counter(participant) - collections.Counter(completion)
    return list(answer.keys())[0]
```

Collections 모듈

{"자료이름" : "개수} 형식으로 만들어 준다.

Counter 객체들끼리 뺄셈 가능

```python
def solution(phone_book):
    for a in phone_book:
        for b in phone_book:
            if a==b: continue
            elif b[:len(a)] == a:
                return False
    return True
```