

# BinarySearch

최적화 문제를 결정 문제로 해결하기

김태현

# 최적화 문제(optimize)

---

특정 조건을 만족하는 최대/최소값을 구하는 문제

1.  $\text{optimize}(A, x)$  : A수열에서  $x$  이상인 가장 작은 값 반환
2.  $\text{optimize}(S)$  :  $1+2+\dots+N \leq S$  를 만족하는 큰  $N$  반환
3.  $\text{optimize}(S)$  :  $S \leq N^2$  인 가장 작은  $N$  반환
4.  $\text{optimize}(A, M)$  : A수열을 연속된 부분수열  $M$ 개로 나눈다. 각 부분수열 합의 최대값이 최소가 되게 할때의 최대값 반환

# 결정 문제(decision)

---

true / false 의 답만이 존재하는 문제

1.  $\text{decision}(A, x, y)$  : A수열의 원소  $y$ 가  $x$  이상인가?
2.  $\text{decision}(S, n)$  :  $1+2+\dots+n \leq S$  를 만족하는가?
3.  $\text{decision}(S, n)$  :  $S \leq n^2$  을 만족하는가?
4.  $\text{decision}(A, M, \text{limit})$  : A수열을  $M$ 개로 나눌 때, 각 부분수열의 합이  $\text{limit}$  이하가 되게 만들 수 있는가?

# A 수열에서 x 이상인 가장 작은 값 반환

optimize(A, x) : A수열에서 x 이상인 가장 작은 값 반환

decision(A, x, y) : A수열의 원소 y가 x 이상인가?

**decision(A, 11, y)** : A수열의 원소 y가 11이상인가?

| i   | 0 | 1 | 2 | 3 | 4  | 5  | 6  | 7  | 8  | 9  |
|-----|---|---|---|---|----|----|----|----|----|----|
| y   | 1 | 3 | 4 | 7 | 10 | 13 | 14 | 20 | 25 | 30 |
| o/x | x | x | x | x | x  | o  | o  | o  | o  | o  |

- linear search :  $O(N)$
- binary search :  $O(\log N)$

## 결정문제가 되기 위한 조건

조건의 만족여부 판단이 가능해야 한다.

## binary search가 적용되기 위한 조건

정답(res)을 기준으로 조건의 만족여부가 나뉘어야 한다.

- 최대값을 구하는 경우 : o o o | x x x
- 최소값을 구하는 경우 : x x x | o o o

```
int optimize(A, x)
    s = 0, e = N - 1
    while(s <= e)
        int mid = (s+e)/2
        if(decision(A, x, A[mid])) e = mid - 1
        else s = mid + 1
    return s
```

## 탐색 결과

s : 조건을 만족하는 첫번째 위치

e : 조건을 만족하지 않는 마지막 위치

```
bool decision(A, x, y)
    return y <= x
```

# 1+2+..+N <= S 를 만족하는 큰 N 반환

optimize(S) : 1+2+..+N <= S 를 만족하는 큰 N 반환

decision(S, n) : 1+2+..+n <= S 를 만족하는가?

**decision(14, n)** : 1+2+..+n <= 14 를 만족하는가?

| n   | 1 | 2 | 3 | 4  | 5  | 6  | 7  | 8  | 9  |
|-----|---|---|---|----|----|----|----|----|----|
| sum | 1 | 3 | 6 | 10 | 15 | 21 | 28 | 36 | 45 |
| o/x | o | o | o | o  | x  | x  | x  | x  | x  |

- linear search :  $O(N)$
- binary search :  $O(\log N)$

## 결정문제가 되기 위한 조건

조건의 만족여부 판단이 가능해야 한다.

## binary search가 적용되기 위한 조건

정답(res)을 기준으로 조건의 만족여부가 나뉘어야 한다.

- 최대값을 구하는 경우 : o o o | x x x
- 최소값을 구하는 경우 : x x x | o o o

```
int optimize(S)
```

```
    s = 1, e = S의 범위에 따른 답이 가능한 최대값
```

```
    while(s <= e)
```

```
        int mid = (s+e)/2
```

```
        if(decision(S, mid)) s = mid + 1
```

```
        else e = mid - 1
```

```
    return e
```

## 탐색 결과

s : 조건을 만족하지 않는 첫번째 위치

e : 조건을 만족하는 마지막 위치

```
bool decision(S, n)
```

```
    return (1+n)*n/2 <= S
```

\* for문으로도 가능

**감사합니다**