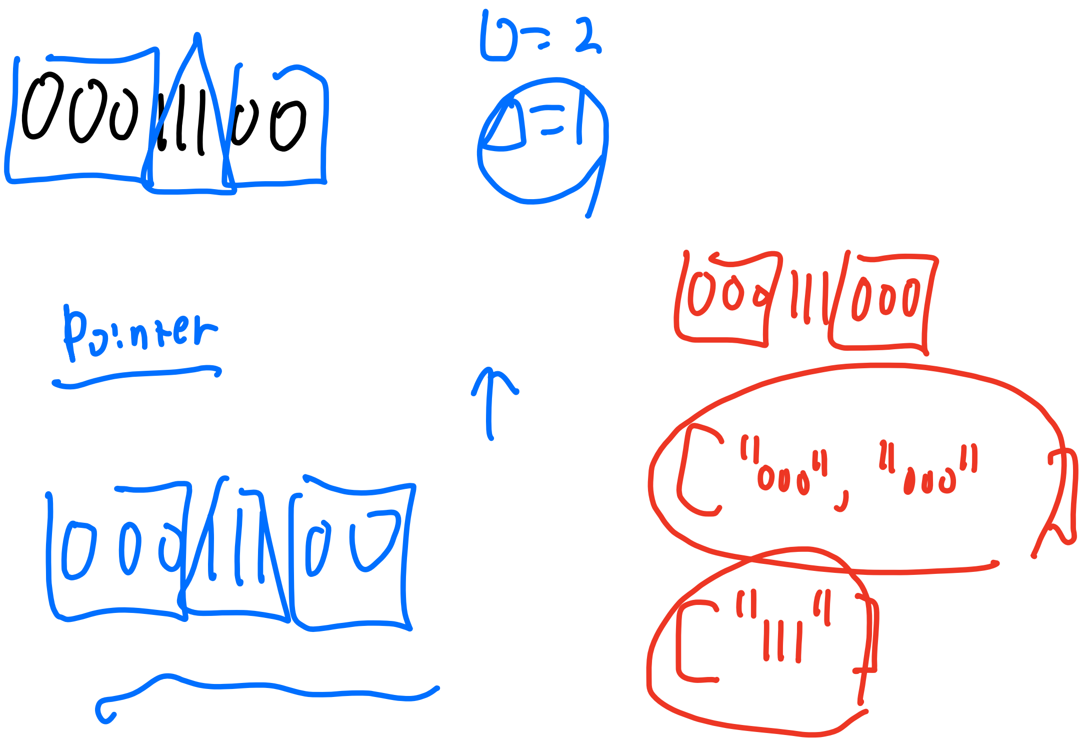
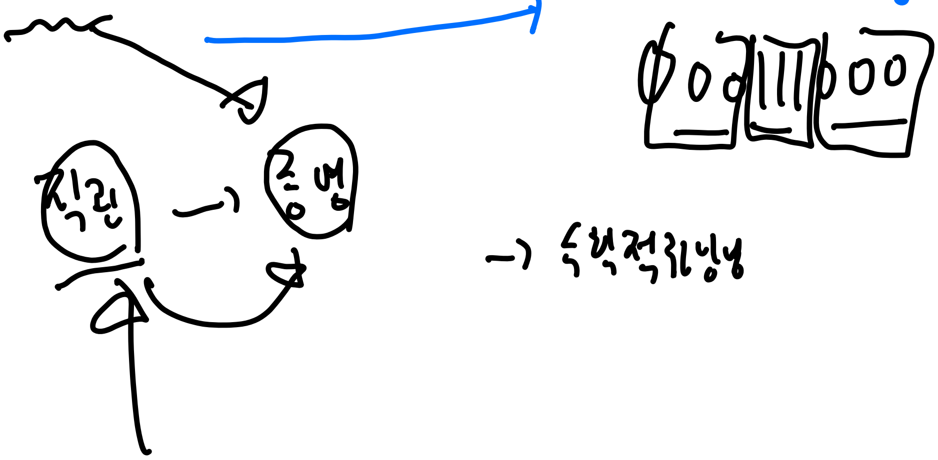


그리디 → 휴리스틱

다익스트라 → 증명



→ 수학적 귀납법

그리디c

→ 탐색공간

→ 가능 → 최소

2^n

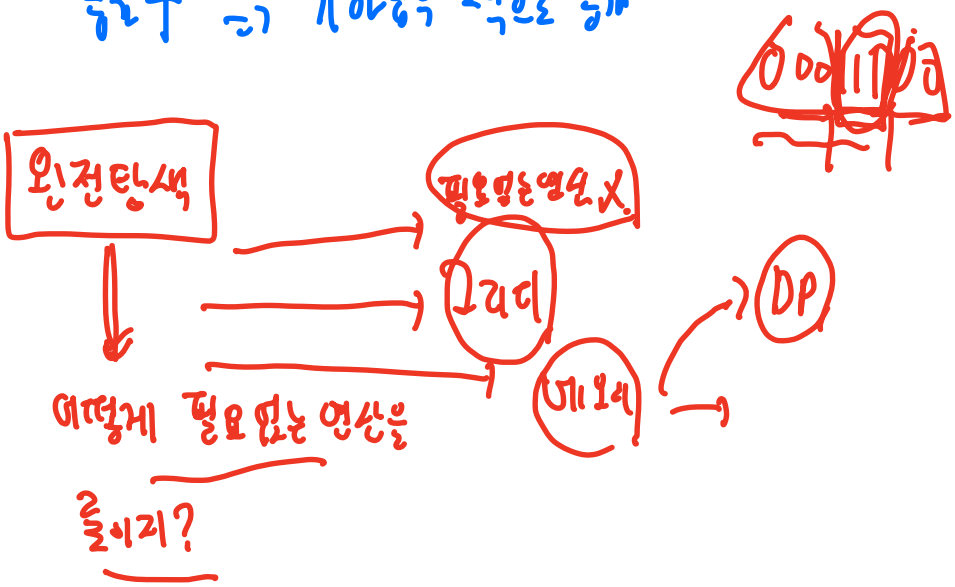
→ 노가다. ⇒ 완전탐색, Brute Force.

$O(2^n)$ ⇒ 쓰는 알고리즘?

$n = 30$

↓

증가 ⇒ 기하급수적으로 증가



수학문제 vs 알고리즘

↓

경우의 수

⇓

노가다 ⇒ $_3C_2$

- 어떻게 효율적으로 노가다

- 노가다 시 ⇒ 어떻게 연산을 줄일것인가?

집합 = { x | x > 0 }

A = { "사과", "바나나", "포도" }

S = { x | x ∈ A }

X = [1, 2, 3, 4, 5, 6].

S = [i for i in X].

[i*2 for i in X].

[int(input()) for _ in range(10)].

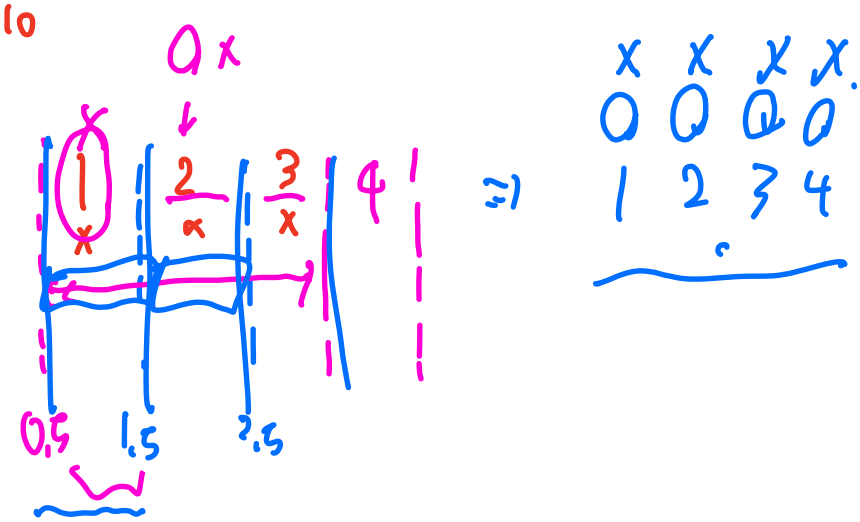
[i*2 for i in X if i%2==0].

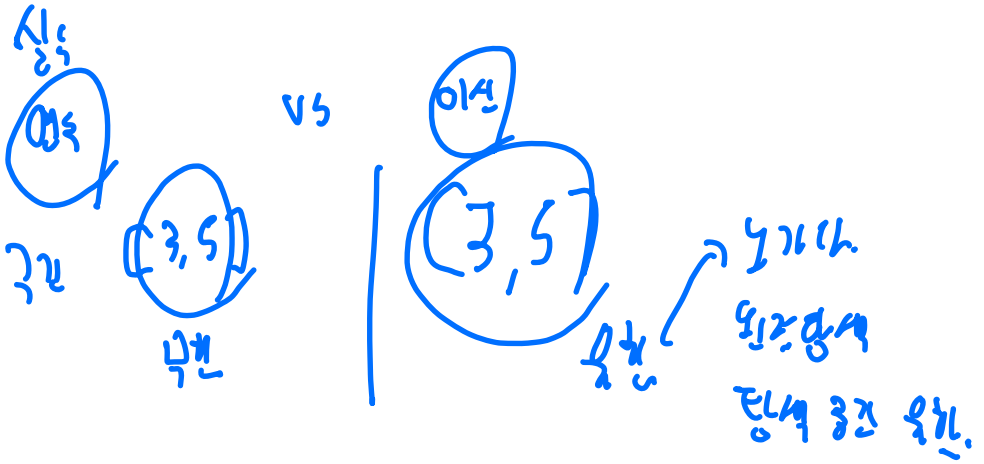
[5, 4, 3, 2, 6, 8, 4 ...]

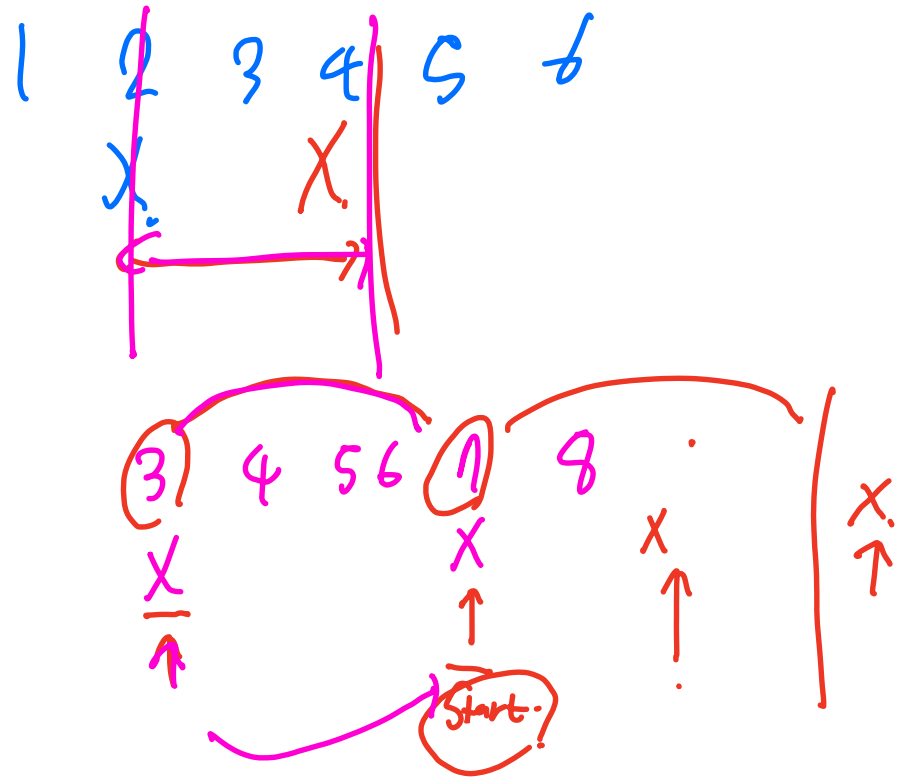
⑤

```
N = int(input())
S = [ int(input()) for _ in range(N) ].
```

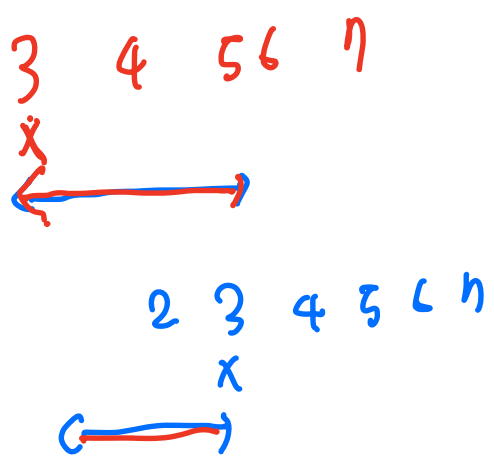
1
2
4
6
10

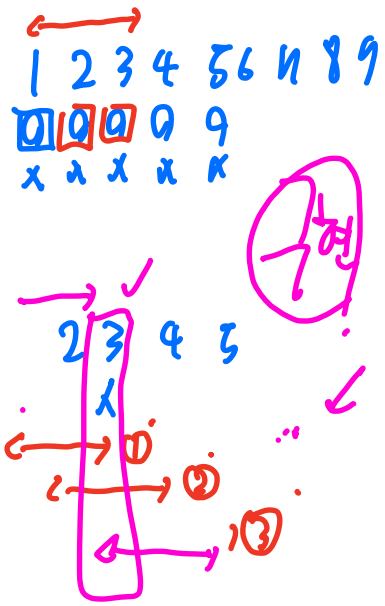






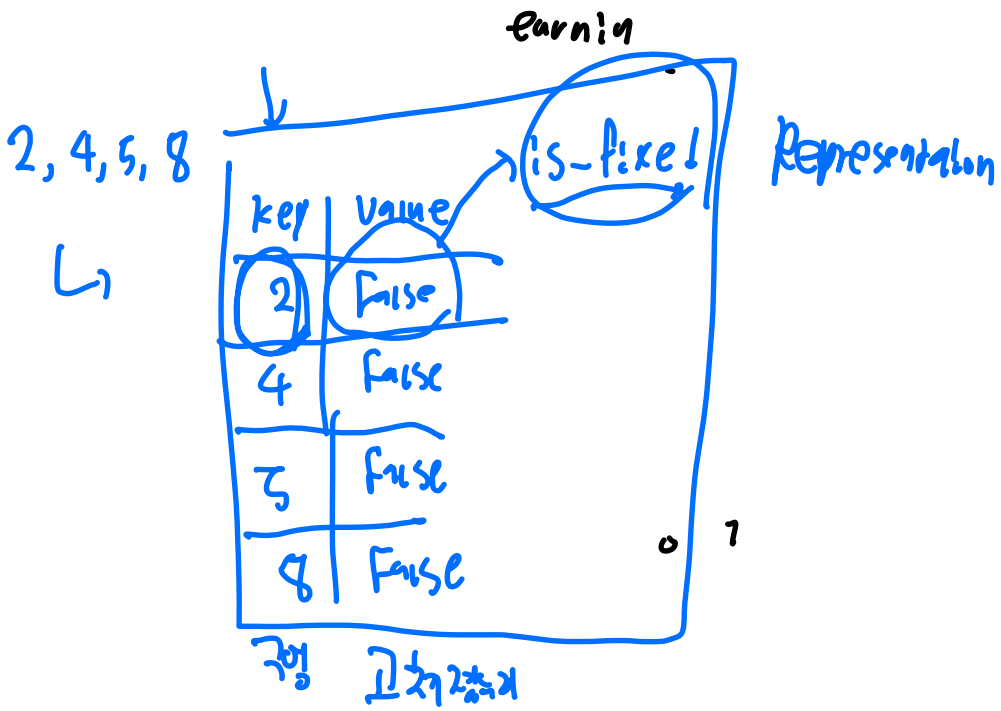
그래서 정답은.

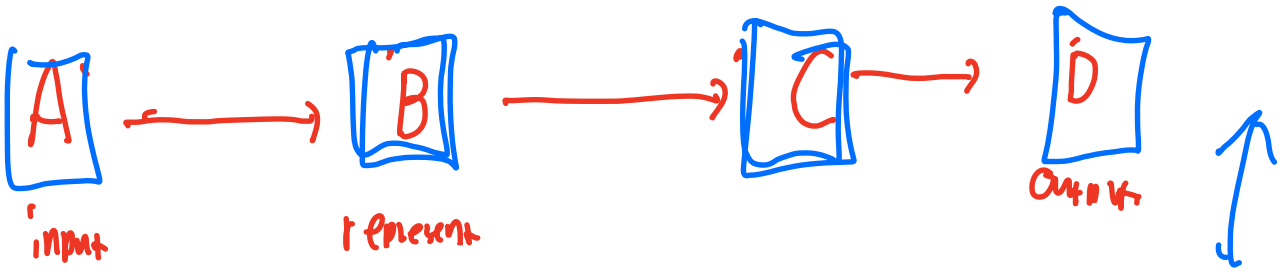


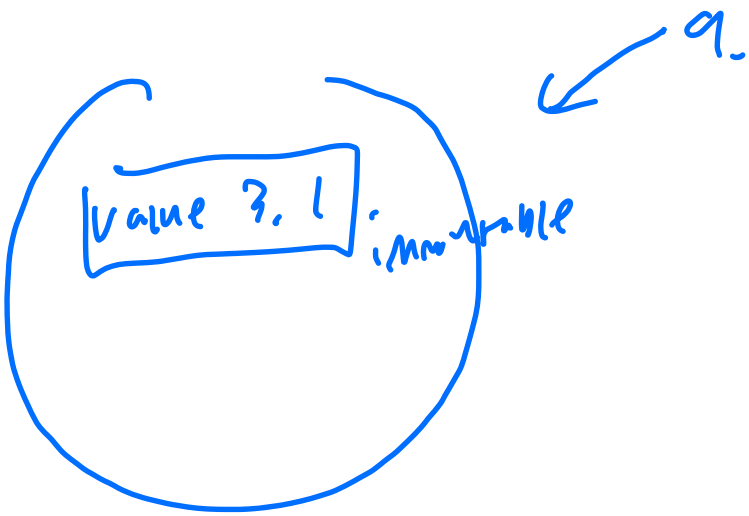


Representation

→ Representation




A
B
C
D


value 3.1