DP

이진탐색 - 파라메트릭 서치

이진탐색

↳ Up down 게임 1 ~ 100

50

=) 정렬 =) 원하는 값을 찾아내기 위해서

일반 선형탐색 O(n)

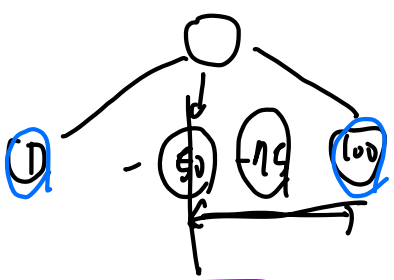
이진탐색 O(logn)

파라메트릭 서치 ⇐

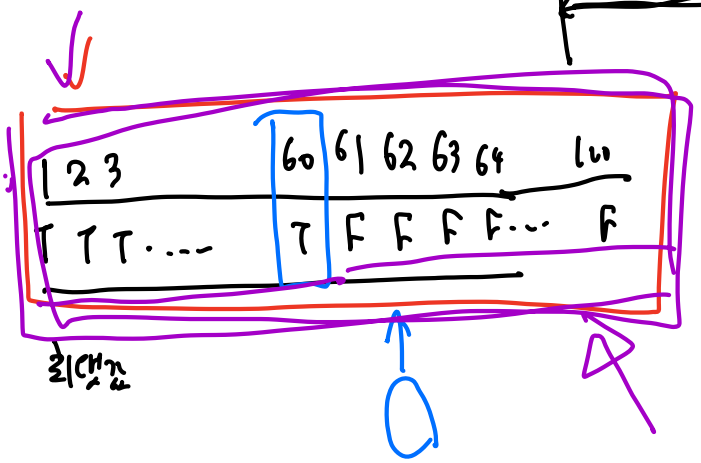
최적화 문제 → 결정문제 → 이진탐색

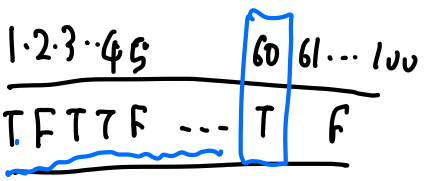
최댓값/최솟값 구하는문제

50 -> 가능? 불가능

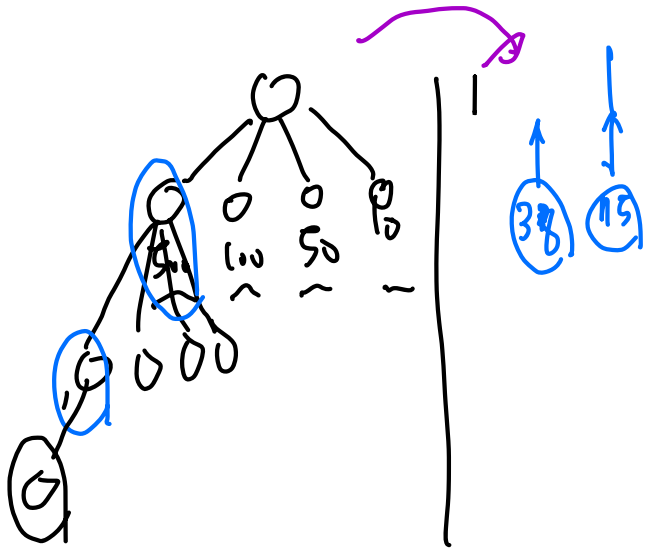


While start <= end:
mid = (start + end) // 2
if 가능:
start = mid+1
else:
end = mid-1





X 파라메트릭 서치

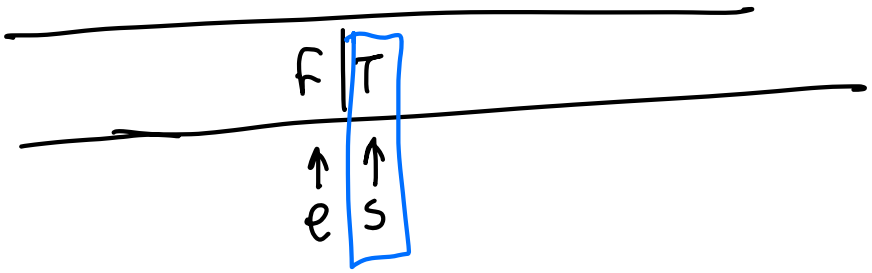


1 2 3 4 5 6 7 8 9
F F F F T T T T F

드래곤 볼 던전

① 파라메트릭 서치 활용하기.

1



던전을 돌때

⇒ 가장 많은 체력 소모가 일어났을 때 x → 정답: x+1.

dp ⇒ i번째 방까지의 최대 체력소모값

dp[i+1] = max(dp[i], 얻은 값)

그리디

for room in rooms:

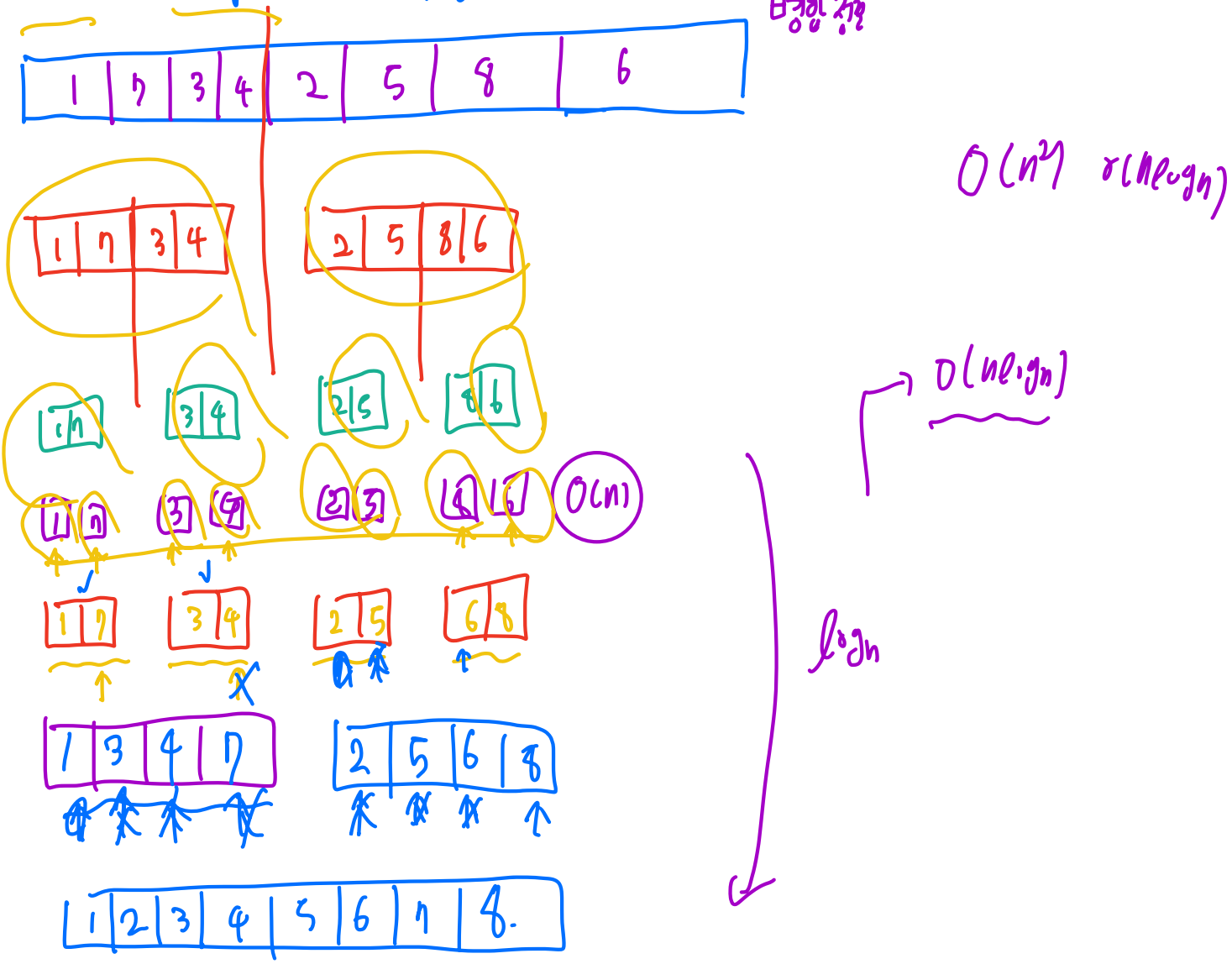
최댓값 갱신

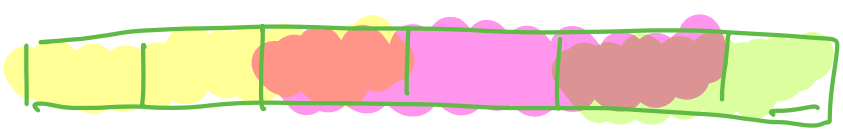
Dynamic Programming.

↳ Top down

↳ Bottom up.

Divide & Conquer. 분할정복 알고리즘





Dynamic programming 은 무엇인가...?!

① 복잡한 문제가 간단해진다

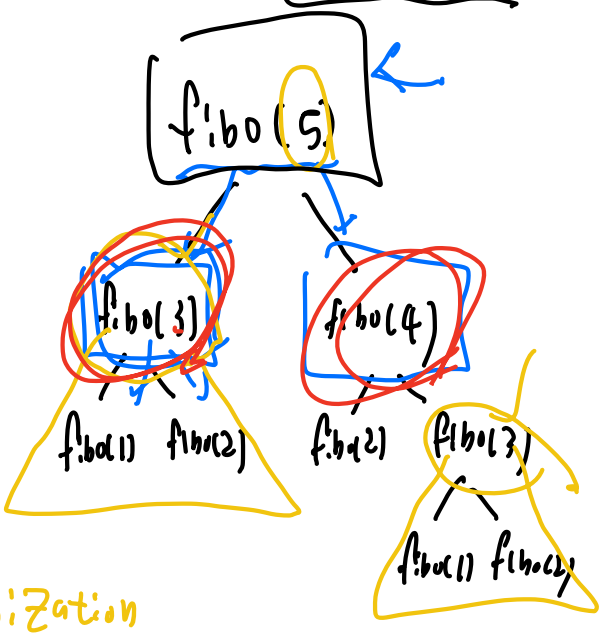
② 복잡한 최적 구조를 만족시킨다

- Top down
- Bottom up

$a_{n+2} = a_{n+1} + a_n$ $a_0 = 1, a_1 = 1$

```
def fibo(n):
    dp = [-1]*(n+1)
    if n==0 or n==1:
        return 1
    elif: dp[n] != -1:
        return dp[n]
    else:
        dp[n] = fibo(n-2) + fibo(n-1)
        return dp[n]
```

Memoization



```
fibo(n):
    dp = [-1] * (n+1)
    dp[0] = 1
    dp[1] = 1
    for i in range(2, n+1):
        dp[i] = dp[i-1] + dp[i-2]
```

tabulation

table

tabular

dp	0	1	2
	.	.	.

Topdown vs Bottom up.

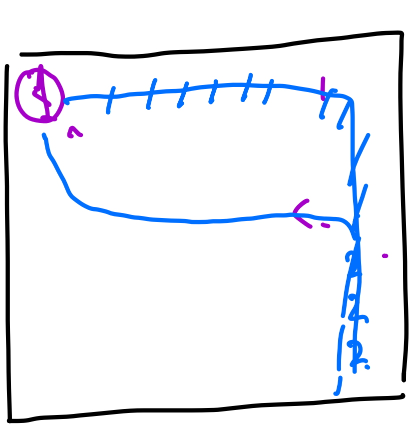
	Topdown	Bottom up
ㅇ	필요한 것만 계산.	재귀 X
ㅅ	재귀	필요없는 것도 계산

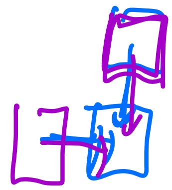
RGB거리

$dp[i][0] = min(dp[i-1][1], dp[i-1][2])$ + red 비용

i번째 집을 Red로 칠할때 최소 비용

내리막길

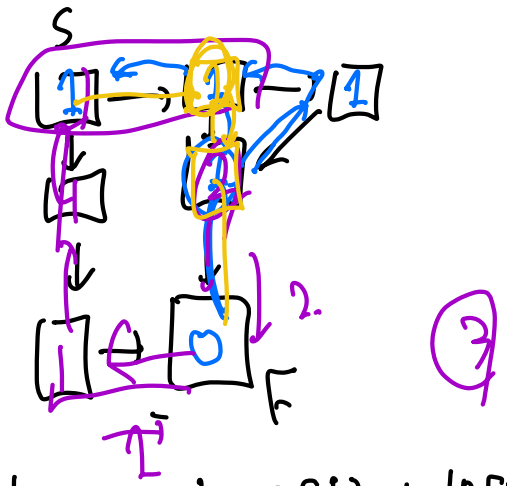




$dp[i][j]$

(i, j)까지 도달하는 경로의 개수

$dp[i][j] = dp[i-1][j] + dp[i][j-1]$



$dp[i][j] = dp[i+1][j] + dp[i][j+1] + \dots$

Knapsack 문제

아이템 5개 → 아이템 무게 가치 (담을 수 있는 무게 10.)

1	10	5
2	30	7
3	20	6
4	15	7
5	10	5

$dp[i][w]$ = Max(안 담았을 때 : $dp[i-1][w]$, 담았을 때 : $dp[i-1][w-w_i] + v_i$)

- $dp[i][w]$: i번째 아이템까지 고려했을 때 최대무게 w일 때 최대가치
- $w - w_i \geq 0$

$dp[1][0] = 0$ $dp[1][1] = 0 \dots dp[1][w_1] = v_1 \dots$

$dp[5][0] \geq 0$

기업투자.

	A	B
1.	10	12
2.	13	16
3.	20	19
4.	21	20
5.	25	21

dp[i][w] max (금액, 2)

= (dp[i-1][w]

dp[i-1][w-1] + v_{i1}

dp[i-1][w-2] + v_{i2}

⋮

dp[i-1][w-5] + v_{i5})

w \ i	0	1	2	…	6
0					
1					
2					

1 3 4

dp[i][w] = (x, y)

x는 i 따졌을때 w 최대금액 → 최대금액

y i번째 얼마투자 했는지

dp[6][10] = (35, 2)

↓

dp[5][8] = (23, 4)

↓

dp[4][4] = (15, 1)

↓

dp[3][3]

다익스트라

dp[노드번호] := 현재노드번호까지의 최단거리

dp[3] = min(dp[0→1→3], dp[0→2→3], dp[0→4→5→3] ...)

5 - 최대 힙

힙기

	0	1	2
노드의 값	a		
자식 노드	3		
비교한 노드	8		