



# Algorithm Week9

---

15기 정규세션

TOBIG'S 14기 정세영

마지막이야 ..



# Contents

---



15기 정규세션  
TOBIG'S 14기 정세영

---

Unit 01 | 7주차 문제 리뷰

---

Unit 02 | 탐욕 알고리즘

---

Unit 03 | 9주차 문제 소개

---



15기 정규세션  
TOBIG'S 14기 정세영

# Unit 01 | 7주차 문제 리뷰

## Problem 1. pooling

```
1  def solve(arr):
2      n = len(arr)
3      inter = []
4      for i in range(0, n, 2):
5          row2 = arr[i:i+2]
6          tmp = []
7          for j in range(0, n, 2):
8              current = row2[0][j:j+2]
9              current.extend(row2[1][j:j+2])
10             current.sort()
11             tmp.append(current[-2])
12         inter.append(tmp)
13     return inter
14
15
16     N = int(input())
17     mat = [list(map(int, input().split())) for _ in range(N)]
18     result = solve(mat)
19     while len(result) != 1:
20         result = solve(result)
21     print(result[0][0])
```

```
1  N = int(input())
2  matrix = []
3  for i in range(N):
4      matrix.append(list(map(int, input().split())))
5
6  def pooling(matrix):
7      if len(matrix) == 2:
8          return sorted(sum(matrix, []))[-2]
9
10     quadrant1 = [m[:len(matrix)//2] for m in matrix[:len(matrix)//2]]
11     quadrant2 = [m[len(matrix)//2:] for m in matrix[:len(matrix)//2]]
12     quadrant3 = [m[:len(matrix)//2] for m in matrix[len(matrix)//2:]]
13     quadrant4 = [m[len(matrix)//2:] for m in matrix[len(matrix)//2:]]
14
15     return pooling([[pooling(quadrant1), pooling(quadrant2)], [pooling(quadrant3), pooling(quadrant4)]])
16
17 print(pooling(matrix))
```

2x2 정사각형으로 나눠준 후 2번째로 큰 수만 살린다.

정사각형 나누는 접근을 아예 재귀로 해도 되고,

반복문을 사용하여 재귀의 비효율적인 측면을 해소할 수도 있다.

# Unit 01 | 7주차 문제 리뷰



15기 정규세션  
TOBIG'S 14기 정세영

## Problem 2. 교양 수업은 재미없다구

```
1 N = int(input())
2 paper = []
3 for i in range(N):
4     paper.append(list(map(int, input().split())))
5
6 def quadrant(P):
7     global white, blue
8     if sum([sum(p) for p in P]) == 0:
9         white += 1
10    elif sum([sum(p) for p in P]) == len(P)**2:
11        blue += 1
12    else:
13        quadrant([p[:len(P)//2] for p in P[:len(P)//2]])
14        quadrant([p[len(P)//2:] for p in P[:len(P)//2]])
15        quadrant([p[:len(P)//2] for p in P[len(P)//2:]]
16        quadrant([p[len(P)//2:] for p in P[len(P)//2:]]
17
18 white, blue = 0, 0
19 quadrant(paper)
20 print(white)
21 print(blue)
```

1번 문제와 유사하게 사분할해가면 된다.

~~numpy를 사용하면 이보다 숏코딩이 가능하지만,  
백준에서는 numpy 못씀---~~



## Problem 3. 회장님의 프로포즈

히스토그램에서 가장 넓이가 큰 직사각형을 찾는 문제!  
왼쪽, 오른쪽으로 쪼개서 탐색하되

1. 왼쪽에서 가장 넓은 직사각형
2. 오른쪽에서 가장 넓은 직사각형
3. 중간에 걸쳐서 가장 넓은 직사각형

나누어 접근하면 된다.

```
1 def big_square(start, end):
2     global hist
3
4     if end-start == 1:
5         answer = hist[start]
6         return answer
7
8     half = int((start + end)/2)
9     left = big_square(start, half)
10    midl, midr, width = half-1, half, 2
11    mid = min(hist[midl], hist[midr]) * width
12
13    while True:
14        if midl-1 < start:
15            break
16        midl -= 1
17        width += 1
18        new_mid = min(hist[midl:midr+1]) * width
19        if mid > new_mid:
20            midl += 1
21            width -= 1
22            break
23        mid = new_mid
24
25    while True:
26        if midr+1 >= end:
27            break
28        midr += 1
29        width += 1
30        new_mid = min(hist[midl:midr+1]) * width
31        if mid > new_mid:
32            midr -= 1
33            width -= 2
34            break
35        mid = new_mid
36
37    right = big_square(half, end)
38
39    answer = max([left, right, mid])
40    return answer
```



15기 정규세션  
TOBIG'S 14기 정세영

## Unit 02 | 탐욕 알고리즘



## 15기 알고리즘 커리큘럼

1주차	OT & 알고리즘 기초
3주차	완전탐색
5주차	동적계획
7주차	분할정복
9주차	탐욕 알고리즘





## 탐욕 알고리즘 (Greedy 알고리즘)

최적해를 구하고 싶을 때 사용하는 알고리즘

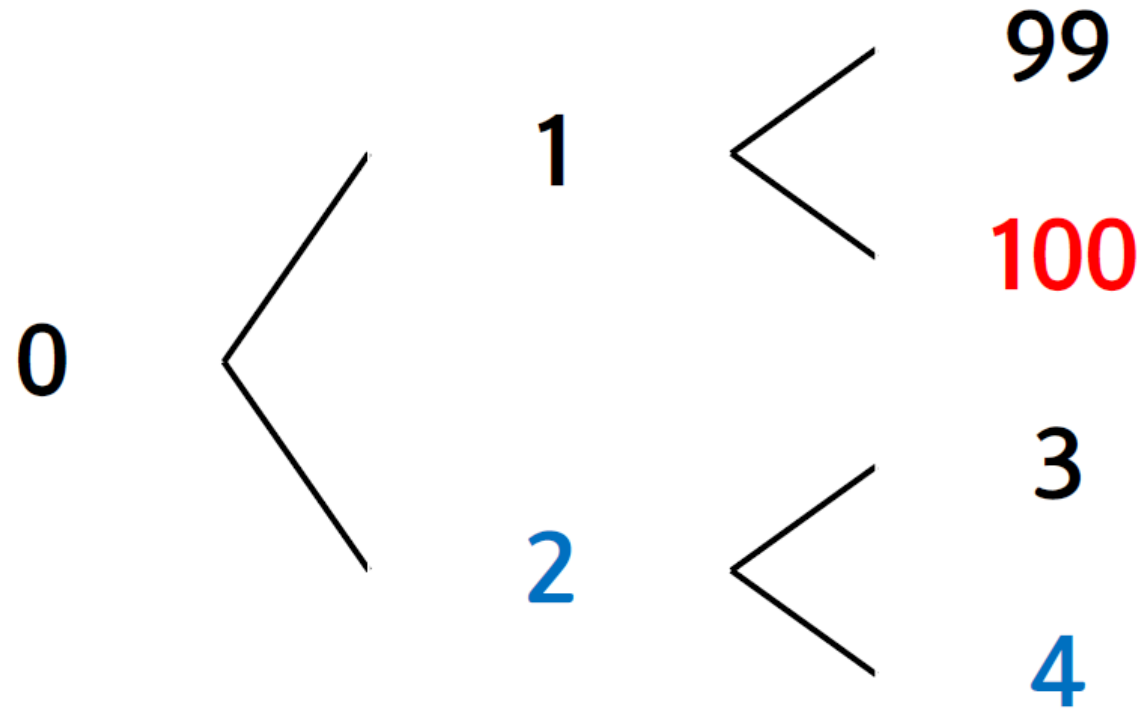
매 상황에서 가장 좋다고 생각하는 것을 선택해 나가자!

가장 좋은 결과를 얻는 것이 보장되는 것은 아니지만, 계산 속도가 매우 빠름



그리디 문제를 풀 때 과도한 재귀나 반복문을 사용하면  
시간초과가 날 수 있다는 얘기!

## 탐욕 알고리즘 (Greedy 알고리즘)





**Q.** 거스름돈 9430원을 동전(지폐)의 수를 가장 적게 주는 방법은?

**A.** 큰 단위부터 거슬러주자!

$$1 * 5000 + 4430$$

$$1 * 5000 + 4 * 1000 + 430$$

$$1 * 5000 + 4 * 1000 + 4 * 100 + 30$$

$$1 * 5000 + 4 * 1000 + 4 * 100 + 3 * 10$$

$$> 12$$



15기 정규세션  
TOBIG'S 14기 정세영

## Unit 03 | 9주차 문제 소개

## Problem 1. 운영체제 학점 잘 받음?

의정이의 컴퓨터는 1개의 CPU만을 가지고 있다. 그리고 이 CPU에는 N개의 프로세스가 대기하고 있다. 각 프로세스는 1번부터 N번까지 번호가 매겨져 있으며, i번 프로세스의 실행 시간은  $P_i$ 분이다.

프로세스들의 실행 순서가 어떻게 스케줄링되는지에 따라서, 각 프로세스가 종료 되기까지 걸리는 시간이 달라진다. 여기서 한 프로세스가 종료 되기까지 걸리는 시간을 Turnaround Time이라고 부른다.

예를들어  $P_1 = 3$ ,  $P_2 = 1$ ,  $P_3 = 4$ ,  $P_4 = 3$ ,  $P_5 = 2$  인 경우에 [1, 2, 3, 4, 5] 순서로 스케줄링 한다면, 1번 프로세스의 Turnaround Time은 3분이다. 2번 프로세스는 1번 프로세스가 실행되는 동안 기다려야 하기 때문에, Turnaround Time은  $3+1 = 4$ 분이 된다. 3번 프로세스는 1번, 2번프로세스가 실행되는 동안 기다려야 하기 때문에, 총  $3+1+4 = 8$ 분이 필요하게 된다. 4번 프로세스는  $3+1+4+3 = 11$ 분, 5번 프로세스는  $3+1+4+3+2 = 13$ 분이 걸리게 된다.

이 경우에 각 프로세스의 Turnaround Time의 합은  $3+4+8+11+13 = 39$ 분이 된다.

대기하고 있는 프로세스의 수 N과 각 프로세스가 실행되는데 걸리는 시간  $P_i$ 가 주어졌을 때, Turnaround Time 합의 최솟값을 구해보자!



## Problem 1. 운영체제 학점 잘 받음?

입력으로 첫째 줄에 프로세스의 수  $N(1 \leq N \leq 1,000)$ 이 주어진다. 둘째 줄에는 각 프로세스의 실행 시간  $P_i$ 가 주어진다. ( $1 \leq P_i \leq 1,000$ )

출력 첫째 줄에 Turnaround Time 합의 최솟값을 출력한다.

입력1

5

3 1 4 3 2

출력1

32

어느컴공과학생이 인문계건물로 침입했다  
경비원은 사람들을 모아놓고 질문을하였다

★ ★ ♥ ★ ★

"하트의 위치는 몇번째인가"

그중한명이 대답하길  
"0... 1...2... 두번째요!"

"재 잡아"

## Problem 2. 라면 물은 생명

완벽한 레시피대로 라면을 끓이고 싶은 지은이는 라면 물을 완벽하게 맞추고 싶다.

하지만 지은이는 5짜리 물병과 3짜리 물병만 가지고 있다.

완벽과 효율을 추구하는 지은이는 물병을 최대한 적게 이용하려고 한다. 예를 들어, 18만큼 물을 냄비에 부어야 할 때, 3짜리로 6번 부어도 되지만, 5짜리 3번과 3짜리 1번을 이용하면 더 적게 물을 부을 수 있다.

지은이가 정확하게  $N$ 만큼 물을 부어야 할 때, 물병을 몇 번 이용하면 되는지 그 수를 구해보자!

## Problem 2. 라면 물은 생명

첫째 줄에 라면 물  $N$ 이 주어진다. ( $3 \leq N \leq 5000$ )

지은이의 최소 물병 이용 횟수를 출력한다. 만약 정확하게  $N$ 을 만들 수 없다면 -1을 출력한다.

예제 입력

18

예제 출력

4







## Problem 3. 투빅코인

원도는 요즘 투빅코인에 빠져있다. 그는 미래를 내다보는 눈이 뛰어나, 날 별로 코인의 가격을 예상하고 언제나 그게 맞아 떨어진다. 매일 그는 아래 세 가지 중 한 행동을 한다.

- 1.코인을 하나 산다.
- 2.원하는 만큼 가지고 있는 코인을 판다.
- 3.아무것도 안한다.

원도는 미래를 예상하는 뛰어난 안목을 가졌지만, 어떻게 해야 자신이 최대 이익을 얻을 수 있는지 모른다. 따라서 당신에게 날 별로 코인의 가격을 알려주었을 때, 최대 이익이 얼마나 되는지 계산을 해달라고 부탁했다. 예를 들어 날 수가 3일이고 날 별로 코인가격이 10,7,6일 때, 코인가격이 계속 감소하므로 최대 이익은 0이 된다. 그러나 만약 날 별로 코인가격이 3,5,9일 때는 처음 두 날에 코인을 하나씩 사고, 마지막날 다 팔아 버리면 이익이 10이 된다.

## Problem 3. 투빅코인

첫 줄에는 날의 수를 나타내는 자연수  $N$  ( $2 \leq N \leq 1,000,000$ )이 주어지고, 둘째 줄에는 날 별 코인가격을 나타내는  $N$ 개의 자연수들이 공백으로 구분되어 순서대로 주어진다. 날 별 코인가격은 10,000 이하이다. 출력으로 최대 이익을 나타내는 정수 하나를 출력한다.

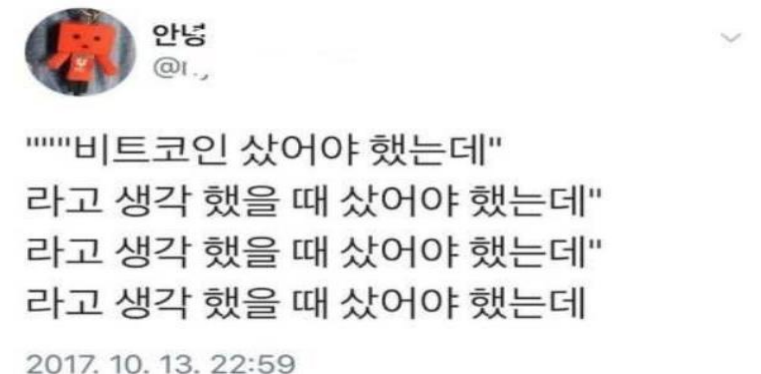
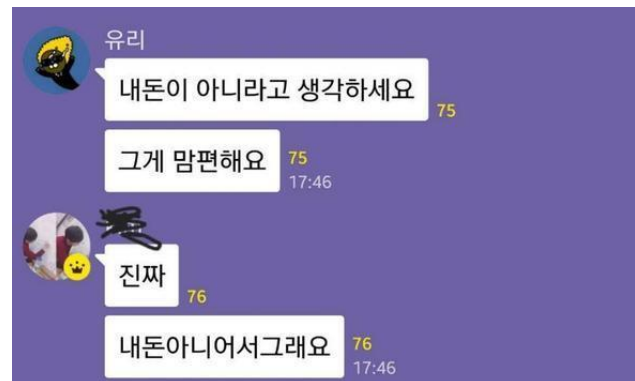
예제 입력

5

1 1 3 1 2

예제 출력

5



# Unit 03 | 9주차 과제 소개

---



15기 정규세션  
TOBIG'S 14기 정세영

코스 코드

0563bdb1ba

코스 링크

<https://class.mimir.io/courses/0563bdb1ba/registrations/new>

둘 중 아무 방식으로 참여하시면 됩니다~!



15기 정규세션  
TOBIG'S 14기 정세영

들어주셔서 감사합니다.

TOBIG'S