

1주차 주간학습보고서(3/22~3/28)

분류 1	전공		
팀 명	정윙팡팡	제출자	이윤희
학습 일시	2021.03.24 15:00 ~ 2021.03.24 17:00	마감 일시	2021.03.24 17:00
제출 일시	2021.03.22 17:27	수정 일시	2021.03.27 12:57

[2021-1 광운스터디그룹]

주간학습보고서

학습주제	그리디 알고리즘
학습장소	기념관 로비
학습방법 및 학습개요	<p>그리디 알고리즘</p> <p>1. 그리디 알고리즘의 개념</p> <p>2. 문제 (백준 알고리즘 : https://www.acmicpc.net/)</p> <p>1) 5585_거스름돈</p> <p>2) 2810_컵홀더</p> <p>3) 2217_로프</p> <p>4) 1541_잃어버린 괄호</p> <p>5) 1946_신입사원</p> <p>6) 1744_수묶기</p> <p>3. 정리</p> <p>깃허브 인증</p>
다음 주 계획	<p>주제: 다이나믹 프로그래밍 정렬</p> <p>일시: 2021.03.29</p> <p>장소: 기념관 2층</p> <p>보고서 제출자: 최수지</p>

* 학습개요의 내용을 구체적으로 작성해 주세요.

<1주차 학습내용>

1. 그리디 알고리즘의 개념

Greedy Algorithms(탐욕법, 탐욕 알고리즘) **Greedy Algorithm**은 문제를 해결하는 과정에서 그 순간순간마다 최적이라고 생각되는 결정을 하는 방식으로 진행하여 최종 해답에 도달하는 문제 해결 방식이다.

2. 문제

1) 5585번 거스름돈

거스름돈

출처 다국어 분류

Bronze II
난이도 제공: solved.ac

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	128 MB	19570	11737	10104	59.611%

문제

타로는 자주 J이점화점에서 물건을 산다. J이점화점에는 잔돈으로 500엔, 100엔, 50엔, 10엔, 5엔, 1엔이 충분히 있고, 언제나 거스름돈 개수가 가장 적게 잔돈을 준다. 타로가 J이점화점에서 물건을 사고 카운터에서 1000엔 지폐를 한장 냈을 때, 받을 잔돈에 포함된 잔돈의 개수를 구하는 프로그램을 작성하시오.

예를 들어 입력된 예1의 경우에는 아래 그림에서처럼 4개를 출력해야 한다.



- 풀이
(이윤희)

```
1 coin_list = [500,100,50,10,5,1]
2 value = int(input())
3 exchange = 1000 - value
4 total_coin_count = 0
5 for coin in coin_list: # coin list로 인자를 직접 비교하면 됨
6     coin_num = exchange // coin # 몫 연산자
7     total_coin_count += coin_num
8     exchange -= coin_num * coin # 거스름돈을 점점 줄여가는 방법
9 print(total_coin_count)
```

680

5

2) 2810_컵홀더

컵홀더

출처 다국어 분류

Bronze II
난이도 제공: solved.ac

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	128 MB	4990	2014	1764	40.000%

문제

십년이면 강산이 변한다.

강산이네 동네에 드디어 극장이 생겼고, 강산이는 극장에 놀러갔다. 매점에서 콜라를 산 뒤, 자리에 앉은 강산이는 큰 혼란에 빠졌다. 양쪽 컵홀더를 이미 옆 사람들이 차지했기 때문에 콜라를 쏟을 컵 홀더가 없었기 때문이다. 영화를 보는 내내 콜라를 손에 들고 있던 강산이는 극장에 다시 왔을 때는 꼭 콜라를 컵 홀더에 놓겠다는 다짐을 한 후 집에 돌아갔다.

극장의 한 줄에는 자리가 N개가 있다. 서로 인접한 좌석 사이에는 컵홀더가 하나씩 있고, 양 끝 좌석에는 컵홀더가 하나씩 더 있다. 또, 이 극장에는 커플석이 있다. 커플석 사이에는 컵홀더가 없다.

극장의 한 줄의 정보가 주어진다. 이때, 이 줄에 사람들이 모두 앉았을 때, 컵홀더에 컵을 쏟을 수 있는 최대 사람의 수를 구하는 프로그램을 작성하시오. 모든 사람은 컵을 한 개만 들고 있고, 자신의 좌석의 양 옆에 있는 컵홀더에만 컵을 쏟을 수 있다.

S는 일반 좌석, L은 커플석을 의미하며, L은 항상 두개씩 쌍으로 주어진다.

어떤 좌석의 배치가 SLLLLSLL일때, 컵홀더를 *로 표시하면 아래와 같다.

```
*S*LL*LL*S*S*LL*
```

위의 예에서 적어도 두 명은 컵홀더를 사용할 수 없다.

- 풀이
(이윤희)

```

1  # hint : 커피석은 일반석으로 생각
2
3  # 좌석 수 입력
4  num = int(input())
5
6  # 좌석 배치 입력
7  seats_list = input()
8
9  # 리스트에서 LL의 개수 세기
10 count_LL = seats_list.count("LL")
11
12 max_using_cupholds = num
13 if count_LL >= 2: # LL의 개수가 2이상인 경우,
14     # 최대 사용 커피홀더의 수는 좌석수 - (LL(세트)의 개수 - 1)과 같다
15     max_using_cupholds -= count_LL - 1
16 print(max_using_cupholds)
17
18

```

```

9
LLLLLLSSS
7

```

3) 2217번 로프

로프

분류

4 Silver IV

난이도 제공: solved.ac

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	192 MB	24995	10720	8747	42.835%

문제

$N(1 \leq N \leq 100,000)$ 개의 로프가 있다. 이 로프를 이용하여 이런 저런 물체를 들어올릴 수 있다. 각각의 로프는 그 굵기나 길이가 다르기 때문에 들 수 있는 물체의 중량이 서로 다를 수도 있다.

하지만 여러 개의 로프를 병렬로 연결하면 각각의 로프에 걸리는 중량을 나눌 수 있다. k 개의 로프를 사용하여 중량이 w 인 물체를 들어올릴 때, 각각의 로프에는 모두 고르게 w/k 만큼의 중량이 걸리게 된다.

각 로프들에 대한 정보가 주어졌을 때, 이 로프들을 이용하여 들어올릴 수 있는 물체의 최대 중량을 구해내는 프로그램을 작성하시오. 모든 로프를 사용해야 할 필요는 없으며, 임의로 몇 개의 로프를 골라서 사용해도 된다.

- 풀이

(안희승)

```
# 2217

# N(1 <= N <= 100,000)
n = int(input())

a = []
for i in range(n):
    a.append(int(input()))

a.sort()
Max = float('-inf')

for i in range(len(a)):
    w = a[i] * (len(a) - i)
    if Max < w:
        Max = w

print(Max)
```

4) 1541번 잃어버린 괄호

잃어버린 괄호 분류

 Silver II

난이도 제공: solved.ac

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	128 MB	24353	11557	9367	47.771%

문제

세준이는 양수와 +, -, 그리고 괄호를 가지고 식을 만들었다. 그리고 나서 세준이는 괄호를 모두 지웠다.

그리고 나서 세준이는 괄호를 적절히 쳐서 이 식의 값을 최소로 만들려고 한다.

괄호를 적절히 쳐서 이 식의 값을 최소로 만드는 프로그램을 작성하시오.

입력

첫째 줄에 식이 주어진다. 식은 '0'~'9', '+', 그리고 '-'만으로 이루어져 있고, 가장 처음과 마지막 문자는 숫자이다. 그리고 연속해서 두 개 이상의 연산자가 나타나지 않고, 5자리보다 많이 연속되는 숫자는 없다. 수는 0으로 시작할 수 있다. 입력으로 주어지는 식의 길이는 50보다 작거나 같다.

- 풀이

(이윤희)

```
1 list1 = input().split('-')
2 s = 0
3 list2 = list1[0].split('+')
4 for i in list2:
5     s += int(i)
6 for i in list1[1:]:
7     for j in list
8         s -= int(j)
9
10 print(s)
```

30+10+20
60

- 논의사항

반례의 경우를 잘 찾아서 풀이가 맞는지 확인해야한다.

5) 1946번 신입사원

신입 사원

출처 분류

1 Silver I

난이도 제공: solved.ac

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	256 MB	21754	6778	4992	31.823%

문제

언제나 최고만을 지향하는 굴지의 대기업 진영 주식회사가 신규 사원 채용을 실시한다. 인재 선발 시험은 1차 서류심사와 2차 면접시험으로 이루어진다. 최고만을 지향한다는 기업의 이념에 따라 그들은 최고의 인재들만을 사원으로 선발하고 싶어 한다.

그래서 진영 주식회사는, 다른 모든 지원자와 비교했을 때 서류심사 성적과 면접시험 성적 중 적어도 하나가 다른 지원자보다 떨어지지 않는 자만 선발한다는 원칙을 세웠다. 즉, 어떤 지원자 A의 성적이 다른 어떤 지원자 B의 성적에 비해 서류 심사 결과와 면접 성적이 모두 떨어진다면 A는 결코 선발되지 않는다.

이러한 조건을 만족시키면서, 진영 주식회사가 이번 신규 사원 채용에서 선발할 수 있는 신입사원의 최대 인원수를 구하는 프로그램을 작성하시오.

입력

첫째 줄에는 테스트 케이스의 개수 $T(1 \leq T \leq 20)$ 가 주어진다. 각 테스트 케이스의 첫째 줄에 지원자의 숫자 $N(1 \leq N \leq 100,000)$ 이 주어진다. 둘째 줄부터 N 개 줄에는 각각의 지원자의 서류심사 성적, 면접 성적의 순위가 공백을 사이에 두고 한 줄에 주어진다. 두 성적 순위는 모두 1위부터 N 위까지 동석차 없이 결정된다고 가정한다.

- 풀이
(김도연)

```

1  import sys
2  input = sys.stdin.readline
3  t = int(input())
4  for i in range(t):
5      n = int(input())
6      s = [0 for i in range(n + 1)]
7      for j in range(n):
8          a, b = map(int, input().split())
9          s[a] = b
10     min_n = s[1]
11     cnt = 0
12     for k in range(2, n + 1):
13         if s[k] > min_n:
14             cnt += 1
15         else:
16             min_n = s[k]
17     print(n - cnt)

```

6) [공동문제] 1744번 수묵기

수 묵기 문류

4 Gold IV
난이도 제공: solved.ac

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	128 MB	14111	3851	3117	26.843%

문제

길이가 N 인 수열이 주어졌을 때, 그 수열의 합을 구하려고 한다. 하지만, 그냥 그 수열의 합을 모두 더해서 구하는 것이 아니라, 수열의 두 수를 묶으려고 한다. 어떤 수를 묶으려고 할 때, 위치에 상관없이 묶을 수 있다. 하지만, 같은 위치에 있는 수(자기 자신)를 묶는 것은 불가능하다. 그리고 어떤 수를 묶게 되면, 수열의 합을 구할 때 묶은 수는 서로 곱한 후에 더한다.

예를 들면, 어떤 수열이 $\{0, 1, 2, 4, 3, 5\}$ 일 때, 그냥 이 수열의 합을 구하면 $0+1+2+4+3+5 = 15$ 이다. 하지만, 2와 3을 묶고, 4와 5를 묶게 되면, $0+1+(2*3)+(4*5) = 27$ 이 되어 최대가 된다.

수열의 모든 수는 단 한번만 묶거나, 아니면 묶지 않아야한다.

수열이 주어졌을 때, 수열의 각 수를 적절히 묶었을 때, 그 합이 최대가 되게 하는 프로그램을 작성하시오.

입력

첫째 줄에 수열의 크기 N 이 주어진다. N 은 10,000보다 작은 자연수이다. 둘째 줄부터 N 개의 줄에, 수열의 각 수가 주어진다. 수열의 수는 -10,000보다 크거나 같고, 10,000보다 작거나 같은 정수이다.

- 풀이
(최수지)

```

1  N = int(input())
2
3  zero_count = 0
4  one_count = 0
5  po_li = []
6  ne_li = []
7  result = []
8
9  for x in range(N):
10     num = int(input())
11     if num == 0:
12         zero_count += 1
13     elif num == 1:
14         one_count += 1
15     elif num > 1:
16         po_li.append(num)
17     else:
18         ne_li.append(num)
19
20 po_li.sort(reverse=True)
21 ne_li.sort(reverse=False)
22
23 if len(po_li) != 0:
24     for x in range(1, len(po_li), 2):
25         result.append(po_li[x]*po_li[x-1])
26
27     if len(po_li) % 2 != 0:
28         result.append(po_li[-1])
29 if len(ne_li) != 0:
30     for x in range(1, len(ne_li), 2):
31         result.append(ne_li[x]*ne_li[x-1])
32     if len(ne_li) % 2 != 0:
33         if zero_count == 0:
34             result.append(ne_li[-1])
35
36 print(sum(result)+one_count)

```

(안희승)

```

1  import sys
2  num = int(sys.stdin.readline())
3  # Positive
4  p = []
5  # Negative
6  n = []
7  result = []
8
9  # 음수와 양수를 나눠서 저장
10 for i in range(num):
11     a = int(sys.stdin.readline())
12     if a > 0:
13         p.append(a)
14     else:
15         n.append(a)
16
17 # 각 배열을 절댓값 기준으로 정렬
18 p.sort(reverse=True)
19 n.sort()
20
21 # 1의 경우 곱하는 것보다 더해야 최댓값을 구할 수 있음, 이를 위해 1을 만날때 까지만 곱셈연산
22 idx = 0
23 c = False

```



```
24
25 # 양수 계산
26 for i in range(0, len(p), 2):
27     if len(p) % 2 == 0:
28         if p[i] == 1:
29             idx = i
30             c = True
31             break
32         elif p[i + 1] == 1:
33             idx = i
34             c = True
35             break
36     else:
37         result.append(p[i] * p[i+1])
38         if i+1 == len(p) - 1:
39             break
40 else:
41     try:
42         if p[i] == 1:
43             idx = i
44             c = True
45             break
46         elif p[i + 1] == 1:
47             idx = i
48             c = True
49             break
50     else:
51         result.append(p[i] * p[i + 1])
52 except IndexError:
53     result.append(p[i])
54     break
```

```

55
56 # 10이 있는 경우 더해주기
57 if c:
58     for i in range(idx, len(p)):
59         result.append(p[i])
60
61 # 음수 계산
62 for i in range(0, len(n), 2):
63     if len(n) % 2 == 0:
64         result.append(n[i] * n[i+1])
65         if i+1 == len(n) - 1:
66             break
67     else:
68         try:
69             result.append(n[i] * n[i+1])
70         except IndexError:
71             result.append(n[i])
72             break
73
74 print(sum(result))

```

- 논의사항

4명 모두 코드는 다르게 짰지만 위 문제와 같은 경우, 양수, 음수, 0인 경우 등 case별로 조건을 걸어서 문제를 해결해야한다.

3. 정리

깃허브 인증

2021.03.08 ~

☒ 그리디 알고리즘

공동문제 : 1744(수 묶기) -> 고민 많이 해보기!!

이름	5585	2810	2217	1541	1946	1744
김도연	3/10	3/10	3/11	2/10	3/12	3/15
안희승	3/9	3/9	3/10	3/11	3/12	3/13
최수지	3/9	3/10	3/12	3/12	3/13	3/13
이윤희	3/9	3/10	3/11	3/12	3/14	3/14

* 학습활동 사진(필수)



2021. 03. 24 오프라인 스터디 활동 인증