알고리즘 스터디 3주차

김건우 2022. 10. 04.

백준 14720 우유 축제

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	4458	2663	2357	62.239%

문제

영학이는 딸기우유, 초코우유, 바나나우유를 좋아한다.

입맛이 매우 까다로운 영학이는 자신만의 우유를 마시는 규칙이 있다.

1. 맨 처음에는 딸기우유를 한 팩 마신다.
2. 딸기우유를 한 팩 마신 후에는 초코우유를 한 팩 마신다.
3. 초코우유를 한 팩 마신 후에는 바나나우유를 한 팩 마신다.
4. 바나나우유를 한 팩 마신 후에는 딸기우유를 한 팩 마신다.

영학이는 우유 축제가 열리고 있는 우유거리에 왔다. 우유 거리에는 우유 가게들이 일렬로 늘어서 있다.

영학이는 우유 거리의 시작부터 끝까지 걸으면서 우유를 사먹고자 한다.

각각의 우유 가게는 딸기, 초코, 바나나 중 한 종류의 우유만을 취급한다.

각각의 우유 가게 앞에서, 영학이는 우유를 사마시거나, 사마시지 않는다.

우유거리에는 사람이 많기 때문에 한 번 지나친 우유 가게에는 다시 갈 수 없다.

영학이가 마실 수 있는 우유의 최대 개수를 구하여라.

입력

첫째 줄에 우유 가게의 수 N이 주어진다. ($1 \le N \le 1000$)

둘째 줄에는 우유 가게 정보가 우유 거리의 시작부터 끝까지 순서대로 N개의 정수로 주어진다.

0은 딸기우유만을 파는 가게, 1은 초코우유만을 파는 가게, 2는 바나나우유만을 파는 가게를 뜻하며, 0, 1, 2 외의 정수는 주어지지 않는다.

일반적인 풀이

```
N = int(input())
data = list(map(int, input().split()))
result = 0

for i in range(N):
    if data[i] == result % 3:
        result += 1

print(result)
```

백준 2720
세탁소 사장 동혁

거스름돈의 액수가 주어지면 리암이 줘야할 쿼터(Quarter, $0.25)의 개수, 다임(Dime, $0.10)의 개수, 니켈(Nickel, $0.05)의 개수, 페니(Penny, $0.01)의 개수를 구하는 프로그램을 작성하시오. 거스름돈은 항상 $5.00 이하이고, 손님이 받는 동전의 개수를 최소로 하려고 한다. 예를 들어, $1.24를 거슬러 주어야 한다면, 손님은 4쿼터, 2다임, 0니켈, 4페니를 받게 된다.

입력

첫째 줄에 테스트 케이스의 개수 T가 주어진다. 각 테스트 케이스는 거스름돈 C를 나타내는 정수 하나로 이루어져 있다. C의 단위는 센트이다. (1달러 = 100센트) (1<=C<=500)

출력

각 테스트케이스에 대해 필요한 쿼터의 개수, 다임의 개수, 니켈의 개수, 페니의 개수를 공백으로 구분하여 출력한다.

예제 입력 1 복사

```
3
124
25
194
```

예제 출력 1 복사

```
4 2 0 4
1 0 0 0
7 1 1 4
```

일반적인 풀이

```
coins = [25, 10, 5, 1]

for _ in range(int(input())):
    C = int(input())
    result = [0, 0, 0, 0]
    for idx, coin in enumerate(coins):
        result[idx] += C // coin
        C %= coin

    print(*result)
```

백준 2864
5와 6의 차이

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	128 MB	9026	6617	6031	74.734%

문제

상근이는 2863번에서 표를 너무 열심히 돌린 나머지 5와 6을 헷갈리기 시작했다.

상근이가 숫자 5를 볼 때, 5로 볼 때도 있지만, 6으로 잘못 볼 수도 있고, 6을 볼 때는, 6으로 볼 때도 있지만, 5로 잘못 볼 수도 있다.

두 수 A와 B가 주어졌을 때, 상근이는 이 두 수를 더하려고 한다. 이때, 상근이가 구할 수 있는 두 수의 가능한 합 중, 최솟값과 최댓값을 구해 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 두 정수 A와 B가 주어진다. (1 <= A,B <= 1,000,000)

출력

첫째 줄에 상근이가 구할 수 있는 두 수의 합 중 최솟값과 최댓값을 출력한다.

예제 입력 1 복사

```
11 25
```

예제 출력 1 복사

```
36 37
```

일반적인 풀이

```
coins = [25, 10, 5, 1]

for _ in range(int(input())):
    C = int(input())
    result = [0, 0, 0, 0]
    for idx, coin in enumerate(coins):
        result[idx] += C // coin
        C %= coin

    print(*result)
```

백준 22864 피로도

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	1024 MB	2712	1185	1055	45.240%

문제

하루에 한 시간 단위로 일을 하거나 일을 쉬어도 된다. 하루에 한 시간 일하면 피로도는 A 만큼 쌓이고 일은 B 만큼 처리할 수 있다.

만약에 한 시간을 쉰다면 피로도는 C 만큼 줄어든다. 단, 피로도가 음수로 내려가면 0으로 바뀐다. 당연히 일을 하지 않고 쉬었기 때문에 처리한 일은 없다.

피로도를 최대한 M 을 넘지 않게 일을 하려고 한다. M 를 넘기면 일하는데 번아웃이 와서 이미 했던 일들도 다 던져버리고 일을 그만두게 된다.

번아웃이 되지 않도록 일을 할때 하루에 최대 얼마나 일을 할 수 있는지 구해보자. **하루는 24시간이다.**

입력

첫 번째 줄에 네 정수 A, B, C, M이 공백으로 구분되어 주어진다.

맨 처음 피로도는 **0**이다.

출력

하루에 번 아웃이 되지 않도록 일을 할 때 최대 얼마나 많은 일을 할 수 있는지 출력한다.

제한

- $1 \leq A \leq 1,000,000$
- $1 \leq B \leq 10,000$
- $1 \leq C \leq 10,000$
- $1 \leq M \leq 1,000,000$

일반적인 풀이

```
A, B, C, M = map(int, input().split())
tired, cnt, time = 0, 0, 0

while time < 24:
    if tired + A <= M:
        tired += A
        cnt += B
    else:
        tired = max(tired - C, 0)
    time += 1

print(cnt)
```

백준 2810 컵홀더

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 꽂을 수 있는 최대 사람의 수를 구하는 프로그램을 작성하시오. 모든 사람은 컵을 한 개만 들고 있고, 자신의 좌석의 양 옆에 있는 컵홀더에만 컵을 꽂을 수 있다.

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

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

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

위의 예에서 적어도 두 명은 컵홀더를 사용할 수 없다.

입력

첫째 줄에 좌석의 수 N이 주어진다. (1 ≤ N ≤ 50) 둘째 줄에는 좌석의 정보가 주어진다.

출력

컵을 컵홀더에 놓을 수 있는 최대 사람의 수를 출력한다.

예제 입력 1 복사

```
3
SSS
```

예제 출력 1 복사

```
3
```

일반적인 풀이

```
N = int(input())
seats = input()

if "LL" in seats:
    print(seats.count("S") + seats.count("LL") + 1)
else:
    print(len(seats))
```

백준 4796 캠핑

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	128 MB	21290	8447	7185	39.381%

문제

등산가 김강산은 가족들과 함께 캠핑을 떠났다. 하지만, 캠핑장에는 다음과 같은 경고문이 쓰여 있었다.

캠핑장은 연속하는 20일 중 10일동안만 사용할 수 있습니다.

강산이는 이제 막 28일 휴가를 시작했다. 이번 휴가 기간 동안 강산이는 캠핑장을 며칠동안 사용할 수 있을까?

강산이는 조금 더 일반화해서 문제를 풀려고 한다.

캠핑장을 연속하는 P일 중, L일동안만 사용할 수 있다. 강산이는 이제 막 V일짜리 휴가를 시작했다. 강산이가 캠핑장을 최대 며칠동안 사용할 수 있을까? (1 < L < P < V)

입력

입력은 여러 개의 테스트 케이스로 이루어져 있다. 각 테스트 케이스는 한 줄로 이루어져 있고, L, P, V를 순서대로 포함하고 있다. 모든 입력 정수는 int범위이다. 마지막 줄에는 0이 3개 주어진다.

출력

각 테스트 케이스에 대해서, 강산이가 캠핑장을 최대 며칠동안 사용할 수 있는지 예제 출력처럼 출력한다.

예제 입력 1 복사

```
5 8 20
5 8 17
0 0 0
```

예제 출력 1 복사

```
Case 1: 14
Case 2: 11
```

일반적인 풀이

```
i = 1

while True:
    L, P, V = map(int, input().split())
    if L == P == V == 0:
        break
    if V % P < L:
        print(f"Case {i}: {V // P * L + V % P}")
    else:
        print(f"Case {i}: {V // P * L + L}")
    i += 1
```

백준 1343 폴리오미노

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	128 MB	7489	3915	3316	52.502%

문제

민식이는 다음과 같은 폴리오미노 2개를 무한개만큼 가지고 있다. AAAA와 BB

이제 '.'와 'X'로 이루어진 보드판이 주어졌을 때, 민식이는 겹침없이 'X'를 모두 폴리오미노로 덮으려고 한다. 이때, '.'는 폴리오미노로 덮으면 안 된다.

폴리오미노로 모두 덮은 보드판을 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 보드판이 주어진다. 보드판의 크기는 최대 50이다.

출력

첫째 줄에 사전순으로 가장 앞서는 답을 출력한다. 만약 덮을 수 없으면 -1을 출력한다.

예제 입력 1 복사

```
XXXXXX
```

예제 출력 1 복사

```
AAAABB
```

일반적인 풀이

```
board = input()

ans = []
i = 0
while i < len(board):
    if board[i:i+4] == 'XXXX':
        ans.append('AAAA')
        i += 4
    elif board[i:i+2] == 'XX':
        ans.append('BB')
        i += 2
    elif board[i] == '.':
        ans.append('.')
        i += 1
    else:
        break

ans = ''.join(ans)
if len(board) == len(ans):
    print(ans)
else:
    print(-1)
```

파이썬 문자열 풀이(윤아)

```
board = input()
valid = True
X = board.split('.')
for item in X:
    if len(item) % 2 != 0:
        valid = False
if valid is True:
    board = board.replace('XXXX', 'AAAA')
    board = board.replace('XX', 'BB')
    print(board)
else:
    print(-1)
```

좀 더 파이썬답게

```
S = input()
S = S.replace("XXXX", "AAAA").replace("XX", "BB")

print(-1) if "X" in S else print(S)
```

백준 1417
국회의원 선거

현재 형택구에 나온 국회의원 후보는 N명이다. 다솜이는 이 기계를 이용해서 그 마을의 주민 M명의 마음을 모두 읽었다.

다솜이는 기호 1번이다. 다솜이는 사람들의 마음을 읽어서 자신을 찍지 않으려는 사람을 돈으로 매수해서 국회의원에 당선이 되게 하려고 한다. 다른 모든 사람의 득표수 보다 **많은** 득표수를 가질 때, 그 사람이 국회의원에 당선된다.

예를 들어서, 마음을 읽은 결과 기호 1번이 5표, 기호 2번이 7표, 기호 3번이 7표 라고 한다면, 다솜이는 2번 후보를 찍으려고 하던 사람 1명과, 3번 후보를 찍으려고 하던 사람 1명을 돈으로 매수하면, 국회의원에 당선이 된다.

돈으로 매수한 사람은 반드시 다솜이를 찍는다고 가정한다.

다솜이가 매수해야하는 사람의 최솟값을 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 후보의 수 N이 주어진다. 둘째 줄부터 차례대로 기호 1번을 찍으려고 하는 사람의 수, 기호 2번을 찍으려고 하는 수, 이렇게 총 N개의 줄에 걸쳐 입력이 들어온다. N은 50보다 작거나 같은 자연수이고, 득표수는 100보다 작거나 같은 자연수이다.

출력

첫째 줄에 다솜이가 매수해야 하는 사람의 최솟값을 출력한다.

예제 입력 1 복사

```
3
5
7
7
```

예제 출력 1 복사

```
2
```

일반적인 풀이

```
N = int(input())
vote_1 = int(input())
vote = [int(input()) for _ in range(N - 1)]
count = 0

if N == 1:
    print(0)
else:
    while max(vote) >= vote_1:
        max_val = max(vote)
        max_idx = vote.index(max_val)
        vote_1 += 1
        vote[max_idx] -= 1
        count += 1

    print(count)
```

백준 1439 뒤집기

문제

다솜이는 0과 1로만 이루어진 문자열 S를 가지고 있다. 다솜이는 이 문자열 S에 있는 모든 숫자를 전부 같게 만들려고 한다. 다솜이가 할 수 있는 행동은 S에서 연속된 하나 이상의 숫자를 잡고 모두 뒤집는 것이다. 뒤집는 것은 1을 0으로, 0을 1로 바꾸는 것을 의미한다.

예를 들어 S=0001100 일 때,

1. 전체를 뒤집으면 11100011이 된다.
2. 4번째 문자부터 5번째 문자까지 뒤집으면 1111111이 되어서 2번 만에 모두 같은 숫자로 만들 수 있다.

하지만, 처음부터 4번째 문자부터 5번째 문자까지 문자를 뒤집으면 한 번에 0000000이 되어서 1번 만에 모두 같은 숫자로 만들 수 있다.

문자열 S가 주어졌을 때, 다솜이가 해야하는 행동의 최소 횟수를 출력하시오.

입력

첫째 줄에 문자열 S가 주어진다. S의 길이는 100만보다 작다.

출력

첫째 줄에 다솜이가 해야하는 행동의 최소 횟수를 출력한다.

예제 입력 1 복사

```
0001100
```

예제 출력 1 복사

일반적인 풀이

```
S = input()
cnt = 1
for i in range(1, len(S)):
    if S[i-1] != S[i]:
        cnt += 1
print(cnt // 2)
```

백준 2828
사과 담기 게임

문제

상근이는 오락실에서 바구니를 옮기는 오래된 게임을 한다. 스크린은 N칸으로 나누어져 있다. 스크린의 아래쪽에는 M칸을 차지하는 바구니가 있다. (M<N) 플레이어는 게임을 하는 중에 바구니를 왼쪽이나 오른쪽으로 이동할 수 있다. 하지만, 바구니는 스크린의 경계를 넘어가면 안 된다. 가장 처음에 바구니는 왼쪽 M칸을 차지하고 있다.

스크린의 위에서 사과 여러 개가 떨어진다. 각 사과는 N칸중 한 칸의 상단에서 떨어지기 시작하며, 스크린의 바닥에 닿을때까지 직선으로 떨어진다. 한 사과가 바닥에 닿는 즉시, 다른 사과가 떨어지기 시작한다.

바구니가 사과가 떨어지는 칸을 차지하고 있다면, 바구니는 그 사과가 바닥에 닿을 때, 사과를 담을 수 있다. 상근이는 사과를 모두 담으려고 한다. 이때, 바구니의 이동 거리의 최솟값을 구하는 프로그램을 작성하시오.

입력

첫째 줄에 N과 M이 주어진다. (1 ≤ M < N ≤ 10) 둘째 줄에 떨어지는 사과의 개수 J가 주어진다. (1 ≤ J ≤ 20) 다음 J개 줄에는 사과가 떨어지는 위치가 순서대로 주어진다.

출력

모든 사과를 담기 위해서 바구니가 이동해야 하는 거리의 최솟값을 출력한다.

예제 입력 1 복사

```
5 1
3
1
5
3
```

예제 출력 1 복사

```
6
```

일반적인 풀이

```
N, M = map(int, input().split())
J = int(input())
apples = [int(input()) for _ in range(J)]

start = 1
end = M
distance = 0

for apple in apples:
    if apple < start:
        distance += (start - apple)
        start = apple
        end = apple + M - 1

    elif apple > end:
        distance += (apple - end)
        end = apple
        start = end - M + 1

print(distance)
```

백준 9237 이장님 초대

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	128 MB	4941	3024	2639	63.713%

문제

농부 상근이는 마당에 심기 위한 나무 묘목 n개를 구입했다. 묘목 하나를 심는데 걸리는 시간은 1일이고, 상근이는 각 묘목이 다 자라는데 며칠이 걸리는지 정확하게 알고 있다.

상근이는 마을 이장님을 초대해 자신이 심은 나무를 자랑하려고 한다. 이장님을 실망시키면 안되기 때문에, 모든 나무가 완전히 자란 이후에 이장님을 초대하려고 한다. 즉, 마지막 나무가 다 자란 다음날 이장님을 초대할 것이다.

상근이는 나무를 심는 순서를 신중하게 골라 이장님을 최대한 빨리 초대하려고 한다. 이장님을 며칠에 초대할 수 있을까?

입력

입력은 두 줄로 이루어져 있다. 첫째 줄에는 묘목의 수 N ($1 \le N \le 100,000$)이 주어진다. 둘째 줄에는 각 나무가 다 자라는데 며칠이 걸리는지를 나타낸 t_i가 주어진다. ($1 \le t_i \le 1,000,000$)

출력

첫째 줄에 며칠에 이장님을 초대할 수 있는지 출력한다. 답이 여러 가지인 경우에는 가장 작은 값을 출력한다. 묘목을 구입한 날이 1일이다.

예제 입력 1 복사

```
4
2 3 4 3
```

예제 출력 1 복사

```
7
```

일반적인 풀이

```
N = int(input())
trees = sorted(list(map(int, input().split())), reverse=True)

max_val = -float('inf')
for idx, val in enumerate(trees):
    max_val = max(max_val, val + idx + 1)

print(max_val + 1)
```

백준 1026 보물

문제

옛날 옛적에 수학이 항상 큰 골칫거리였던 나라가 있었다. 이 나라의 국왕 김지민은 다음과 같은 문제를 내고 큰 상금을 걸었다.

길이가 N인 정수 배열 A와 B가 있다. 다음과 같이 함수 S를 정의하자.

$$S = A[0] \times B[0] + \ldots + A[N-1] \times B[N-1]$$

S의 값을 가장 작게 만들기 위해 A의 수를 재배열하자. 단, B에 있는 수는 재배열하면 안 된다.

S의 최솟값을 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 N이 주어진다. 둘째 줄에는 A에 있는 N개의 수가 순서대로 주어지고, 셋째 줄에는 B에 있는 수가 순서대로 주어진다. N은 50보다 작거나 같은 자연수이고, A와 B의 각 원소는 100보다 작거나 같은 음이 아닌 정수이다.

출력

첫째 줄에 S의 최솟값을 출력한다.

예제 입력 1 복사

```
5
1 1 1 6 0
2 7 8 3 1
```

예제 출력 1 복사

```
18
```

일반적인 풀이

```
N = int(input())
A = sorted(list(map(int, input().split())))
B = sorted(list(map(int, input().split())), reverse=True)
s = 0

for x, y in zip(A, B):
    s += (x * y)

print(s)
```

이진 탐색

이진 탐색 알고리즘

- 순차 탐색: 리스트 안에 있는 특정한 데이터를 찾기 위해 앞에서부터 데이터를 하나씩 확인
  - 시간 복잡도: O(N)
- 이진 탐색: 정렬되어 있는 리스트에서 탐색 범위를 절반씩 좁혀가며 데이터를 탐색
  - 시간 복잡도: O(logN)

재귀를 이용한 이진 탐색 알고리즘

```python
def binary_search(array, target, start, end):
    if start > end:
        return None
    mid = (start + end) // 2
    if array[mid] == target:
        return mid
    elif array[mid] > target:
        return binary_search(array, target, start, mid - 1)
    else:
        return binary_search(array, target, mid + 1, end)

n, target = map(int, input().split())
array = list(map(int, input().split()))

result = binary_search(array, target, 0, n - 1)
if result == None:
    print("원소가 존재하지 않습니다.")
else:
    print(result + 1)
```

반복문을 이용한 이진 탐색 알고리즘

```
target = 25
m_list = [30, 94, 27, 92, 21, 37, 25, 47, 25, 53, 98, 19, 32, 32, 7]
length = len(m_list)

m_list.sort()
left = 0
right = length-1

while left<=right:
    mid = (left+right)//2
    if m_list[mid] == target:
        print(mid+1)
        break
    elif m_list[mid]>target:
        right = mid-1
    else :
        left = mid+1
```