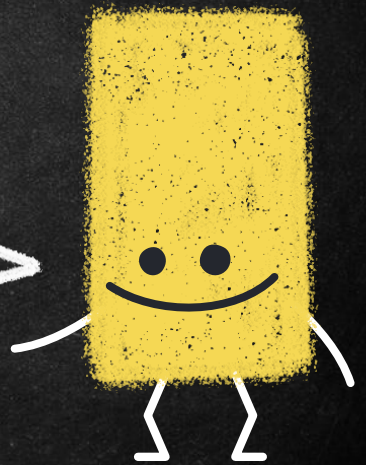
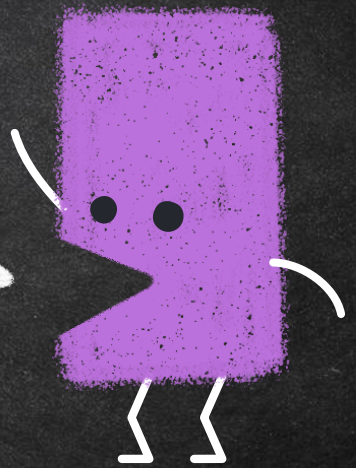
Algorithm Study For 2021

WEEK 6



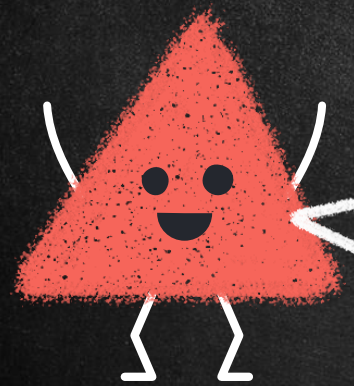
“

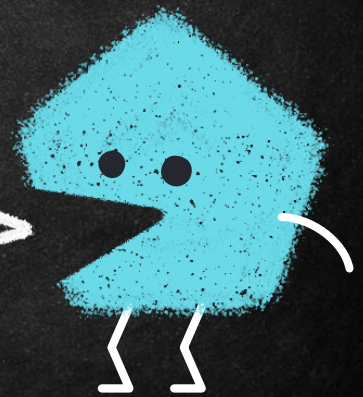
이번 주에는 그리디 알고리즘에
대해서 간단하게 알아봅니다!



1.

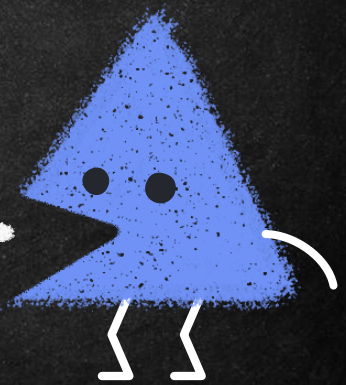
GREEDY !!!





GREEDY ALGORITHM

- 그리디 알고리즘은....
- 당장 눈앞에 보이는 최적의 상황만을 쫓는 알고리즘으로 가장 단순한 형태의 알고리즘입니다!!!
- 항상 최적의 결과를 도출하지는 않지만, 당장 최선처럼 보이는 선택을 하다 보면 어느 정도 좋은 결과를 보장합니다!!!
- 500원 & 100원 & 50원 & 10원의 동전을 가지고 있다고 가정할 때 560원의 거스름돈을 주려면 어떻게 해야할까요...?

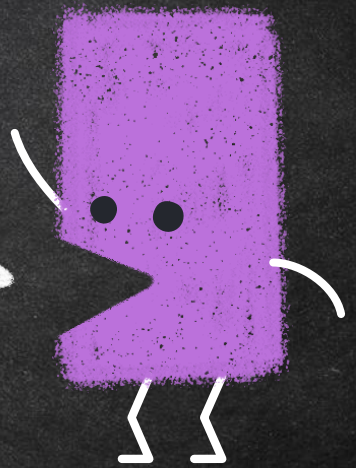


“

예제로 감을 잡아봅시다!

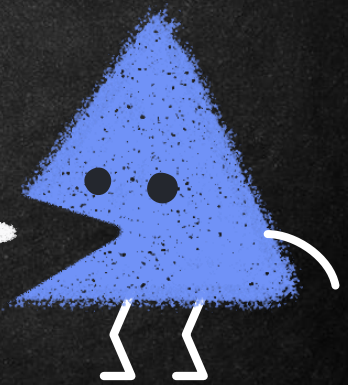
매우 쉬운 거스름돈 문제를 풀어봅시다!!!

문제는 다음과 같습니다...



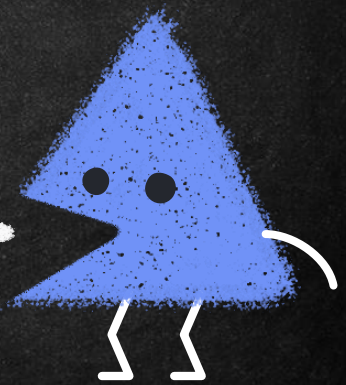
Problem1

거스름돈이 1260원이라고 가정하고, 500원 & 100원 & 50원 & 10원 동전을 이용해서 가장 적은 수의 동전으로 거스름돈을 주려면 몇 개가 필요할까요..?



Solution

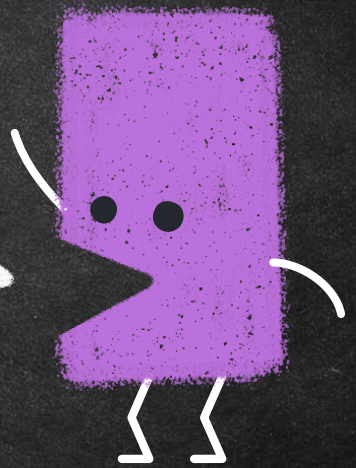
10원짜리 126개를 줄까요..?
50원을 섞어서 줄까요...?
조금만 생각을 해보면...
무조건 큰 숫자의 동전부터 골라서 거슬러주면 됩니다!!!



“

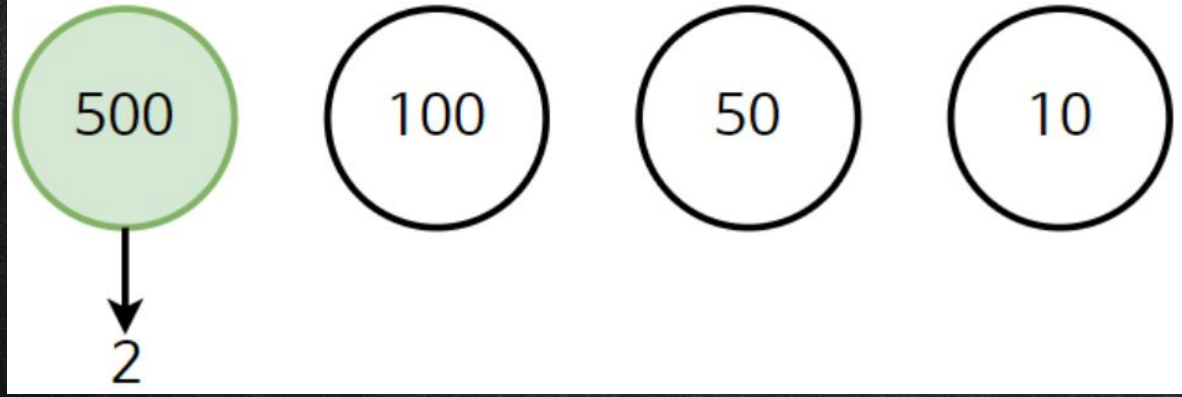
가장 큰 금액인 500원부터
최대한 써봅시다!!!

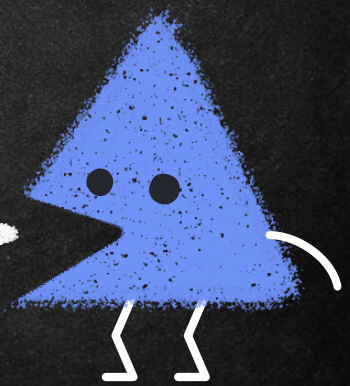
그럼 다음 상황으로 바뀌게
됩니다!!!



Solution

260원을 거슬러주어야 할 때 가장 적은 숫자의 화폐를 이용해 거슬러 주는 경우는?

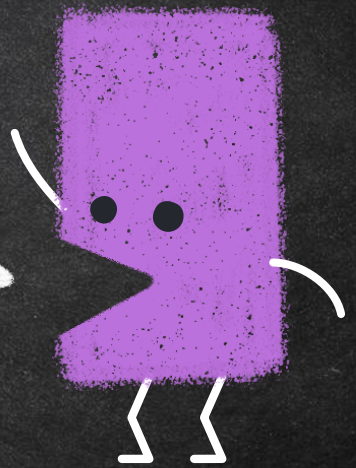




“

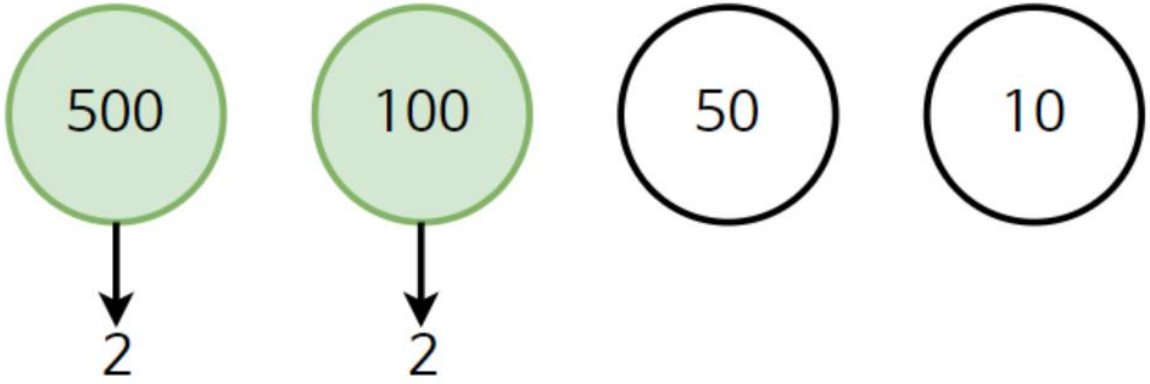
이제 100원짜리를 다 써봅시다!!!

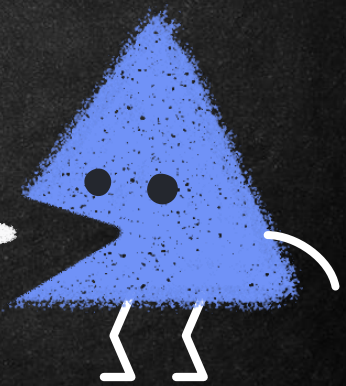
그럼 다음 상황으로 바뀌게 됩니다!!!



Solution

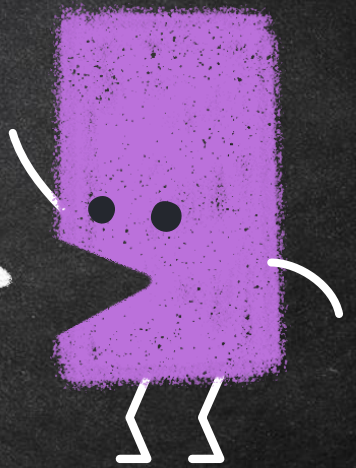
60원을 거슬러주어야 할 때 가장 적은 숫자의 화폐를 이용해 거슬러 주는 경우는?





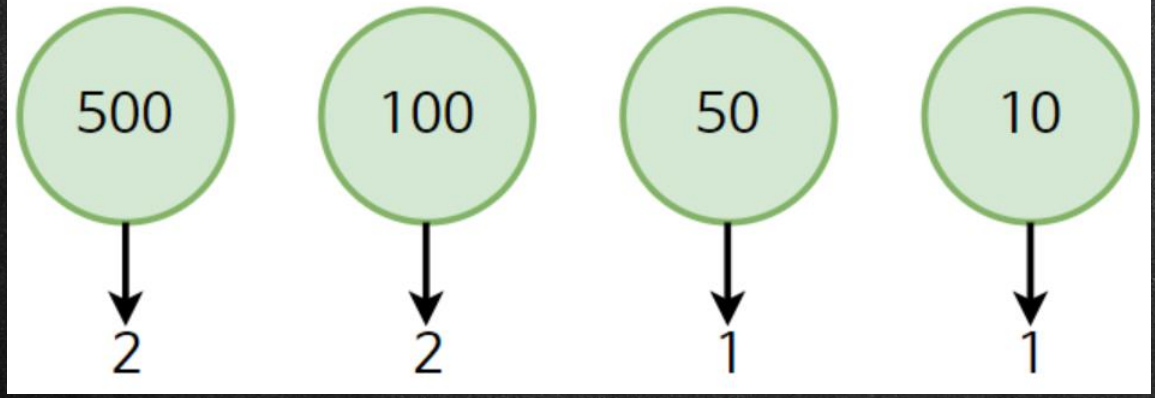
“

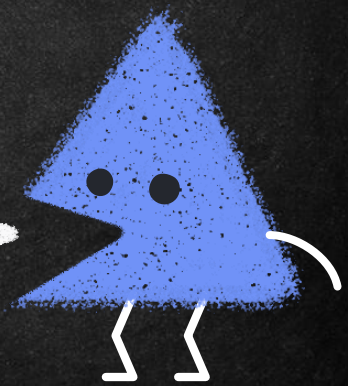
같은 방식으로 0원이 되도록
동전을 추가해줍시다!!!



Solution

0원을 거슬러주어야 할 때 가장 적은 숫자의 화폐를 이용해 거슬러 주는 경우는?

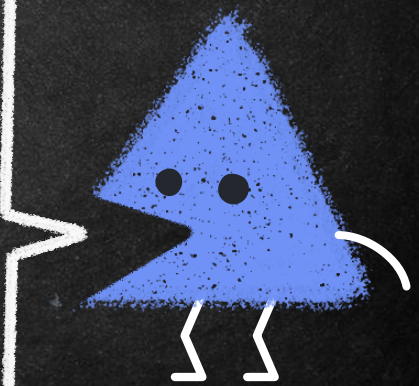




SOURCE CODE

```cpp
#include <iostream>

using namespace std;

int main(void) {
    int n, result = 0;
    cin >> n;
    result += n / 500;
    n %= 500;
    result += n / 100;
    n %= 100;
    result += n / 50;
    n %= 50;
    result += n / 10;
    cout << result;
    return 0;
}
```

```
1260
6
--------------------------------

Process exited after 0.648
계속하려면 아무 키나 누르십
```

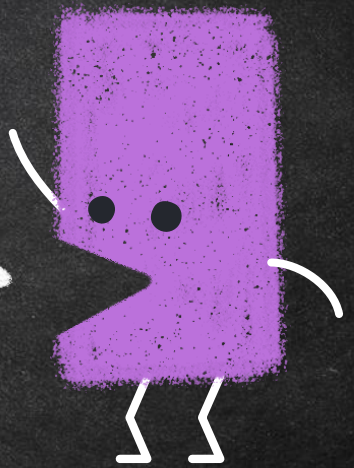


“

약간 어려운 문제를
풀어보겠습니다...

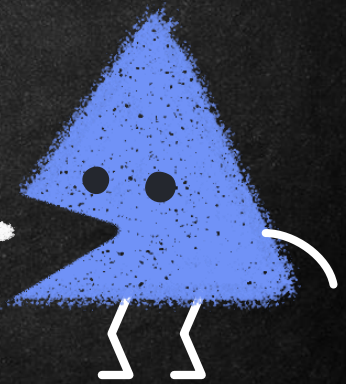
문제는 다음과 같습니다.

출처: 프로그래머스



Problem2

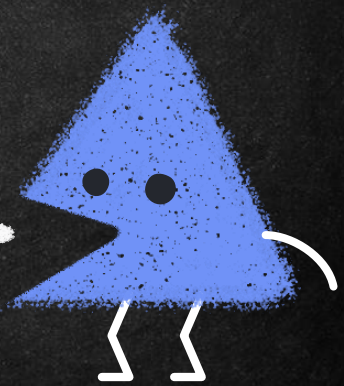
점심시간에 도둑이 들어, 일부 학생이 체육복을 도난당했습니다. 다행히 여벌 체육복이 있는 학생이 이들에게 체육복을 빌려주려고 합니다. 학생들의 번호는 체격 순으로 매겨져 있어, 바로 앞번호의 학생이나 바로 뒷번호의 학생에게만 빌려줄 수 있습니다. 체육복이 있는 학생만 수업을 들을 수 있고, 최대한 많은 학생이 체육 수업을 듣고 싶습니다.



Problem2

입출력 예

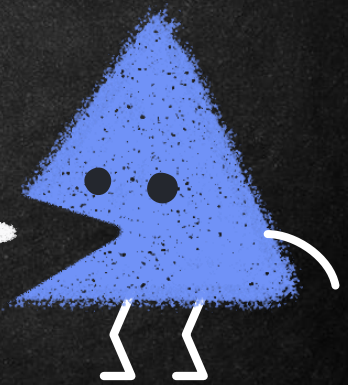
n	lost	reserve	return
5	[2, 4]	[1, 3, 5]	5
5	[2, 4]	[3]	4
3	[3]	[1]	2



Constraints

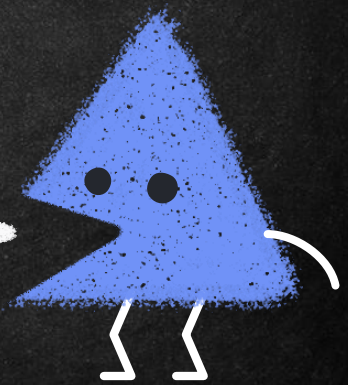
제한사항은 다음과 같습니다.

-여벌 체육복이 있는 학생만 다른 친구에게 빌려줄 수 있습니다.
- 여벌 체육복을 가진 학생이 도둑질을 당할 수도 있습니다. 이때는 1개만 도난을 당한다고 가정합니다.



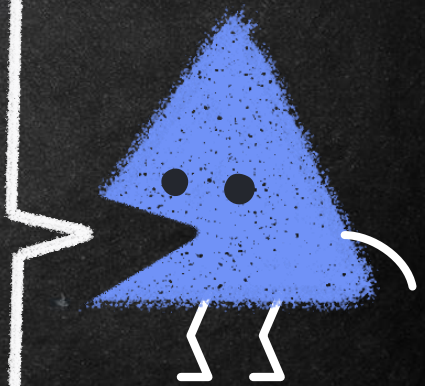
Solution

최적의 해를 찾기 위해서는 어떻게 해야할까요..?
우선, 당장 최선의 선택인 도둑맞은 친구들의 체육복 개수는 -1개를, 여벌이 있는 친구들의 체육복 개수는 +1개를 해준 후에 앞 뒤에 체육복이 2개 있는 친구가 있는지 확인하면 됩니다.



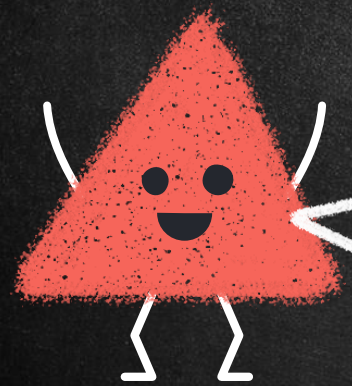
SOURCE CODE

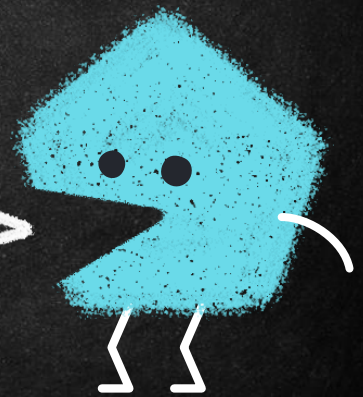
```
int solution(int n, vector<int> lost, vector<int> reserve) {
    vector<int> list(n, 1);
    int answer=0;
    for (int i = 0; i < lost.size(); i++) {
        list[lost[i] - 1]--;
    }
    for (int i = 0; i < reserve.size(); i++) {
        list[reserve[i] - 1]++;
    }
    for (int i = 0; i < list.size(); i++) {
        if (list[i] == 0) {
            if (i != 0 && list[i - 1] == 2) {
                list[i - 1]--;
                list[i]++;
            }
            else if (i != list.size() - 1 && list[i + 1] == 2) {
                list[i + 1]--;
                list[i]++;
            }
        }
    }
    for (int i = 0; i < list.size(); i++) {
        if (list[i] !=0) {
            answer++;
        }
    }
    return answer;
```



2.

PRACTICE!!!



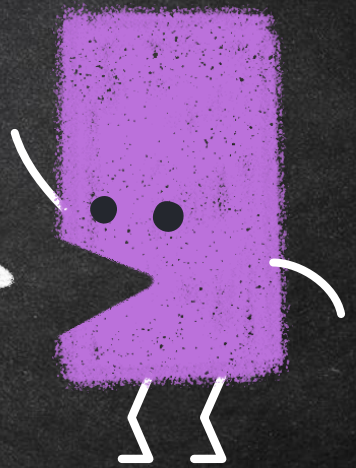


"

다음 문제를 풀어서
제출해주시면 됩니다...

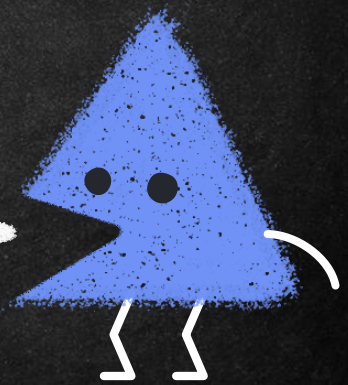
문제는 다음과 같습니다.

출처: 백준



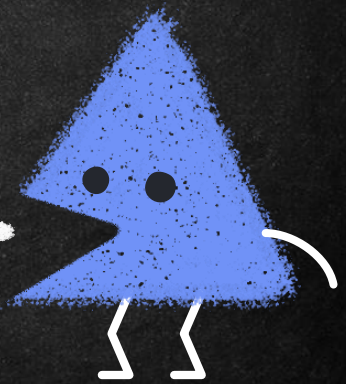
Problem3

https://www.acmicpc.net/problem/11399



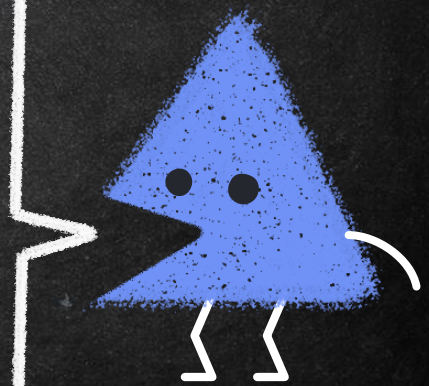
Solution

상황을 이해해보면, 앞서 풀었던 문제들과 크게 다르지 않습니다. 무엇이 최선일까.. 극단적으로 생각을 해보면 됩니다!!!
결국에는 적게 걸리는 사람부터 뽑으면 됩니다!!!
(앞에 사람을 기다리는 시간이 적을수록 자기도 금방 인출이 가능하니까요!!!)



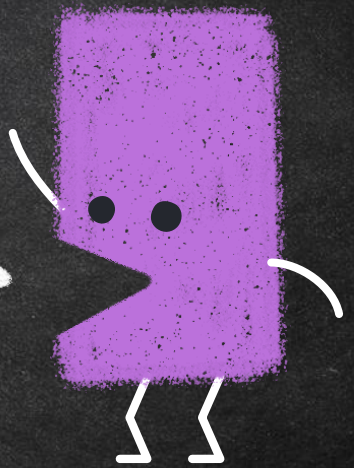
SOURCE CODE

```python
#사람수 입력받기
n=int(input())
#각 사람마다의 시간 입력받기
time=list(map(int,input().split()))

#시간정렬하기
time.sort()
#합산할 변수 초기화
sum1=0
#이전꺼 담을 변수 초기화
prev=0
#각 시간을 이전꺼에 더한것들을 합산하기
for i in time:
  prev+=i
  sum1+=prev

print(sum1)
```



“

파이썬 코드로
제공해드렸습니다. 고민해보신
후에 원하는 언어로 구현하신
후에 제출해주시면 됩니다!!!



THANK YOU