

2023-2024

Soluzioni alla 2° gara



Farmula 1



- [Link al problema](#)

Soluzione

```
points = [0, 25, 18, 15, 12, 10, 8, 6, 4, 2, 1]
input_file = open("input.txt", mode = "r")

t = int(input_file.readline().strip())

for _ in range(t):
    # Leggiamo la riga bianca facendo un readline inutile
    input_file.readline()

    n = int(int(input_file.readline()))

    daniel_score = 0
    max_score = 0
    # Lista con le posizioni di Daniel
    results = list(map(int, input_file.readline().split()))

    # L'idea è calcolare quanti punti ha fatto Daniel e
    # quanti punti ha fatto il primo in classifica
    # Se Daniel è arrivato primo allora prendiamo il secondo classificato
    for i in range(n):
        position = int(results[i])

        # Se Daniel non è arrivato primo
        if position != 1:
            max_score += 25
        else:
            max_score += 18

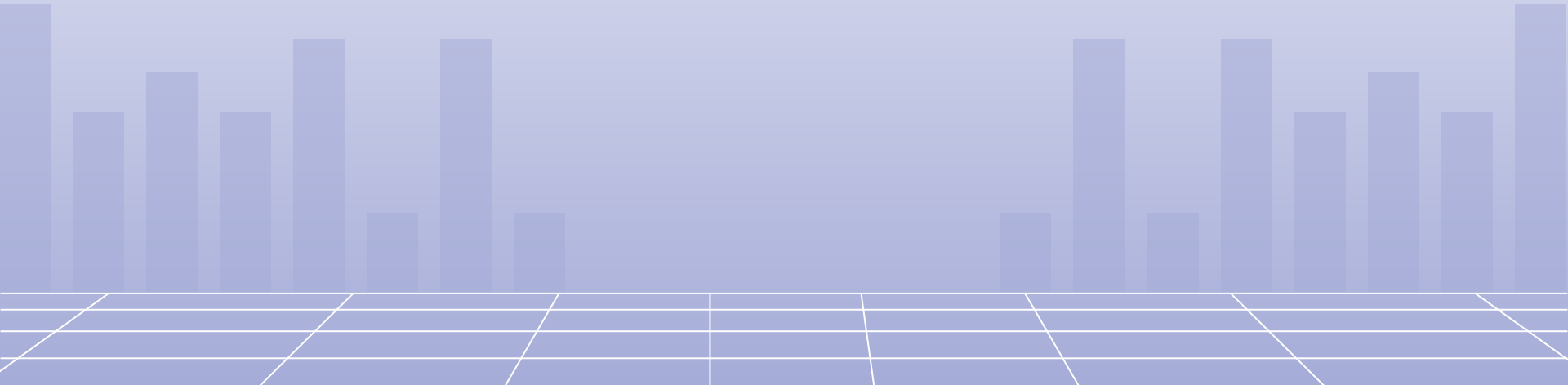
        if position ≤ 10:
            daniel_score += points[position]

    if daniel_score ≥ max_score:
        print("Champion")
    else:
        print("Practice harder")
```

Precise Average 2



- **Link al problema**



Soluzione non molto efficiente...

- Sottraiamo 1\$ a tutti i prodotti (a quelli che possiamo) e vediamo se la media dei prezzi ora è uguale o minore a quella data
- Se la media è ancora maggiore, rifacciamo il punto 1!
- Il problema è che nel caso peggiore dobbiamo sottrarre 1\$ milioni e milioni di volte...
- Possiamo migliorare questo approccio?

```
N, K = map(int, input().split())
K *= N
v = list(map(int, input().split()))

# Il testo ci dice che il prezzo di
# ogni prodotto è massimo 1_000_000_000
max_val = 10**9
sum_val = 0

for x in v:
    sum_val += x

if sum_val == K:
    print(str(0))
else:
    if sum_val < K:
        print(str((K - sum_val + N - 1) // N))
    else:
        i = 1
        while i ≤ max_val:
            sum = 0
            for x in v:
                sum += max(1, x - i)
            if sum ≤ K:
                print(str(i))
                exit()
            i += 1
```

Soluzione ottima

- Abbiamo capito che sottraendo solamente 1\$ ci mettiamo un'eternità...
- E se provassimo a togliere più di un dollaro?
- Dovremmo dimostrare matematicamente qual è la quantità ottimale da togliere ogni volta, potrebbe essere 2, 4, 10, 100, il doppio, il triplo...
- Vi risparmio le dimostrazioni e vi spoilerò che per trovare quel numero nel modo più veloce possibile, raddoppiamo il valore ogni volta. Dovremmo quindi provare a togliere 1\$, poi 2\$, poi 4\$, poi 8\$ e così via....
- Se per esempio togliendo 8\$ a prodotto ci rendiamo conto di aver sorpassato la somma a cui dobbiamo arrivare, dovremmo controllare se per caso potevano andar bene i prezzi da 5\$ a 7\$, visto che non li abbiamo controllati



Soluzione ottima

```
N, K = map(int, input().split())  
K *= N
```

```
sum_val = 0  
v = list(map(int, input().split()))
```

```
for x in v:  
    sum_val += x
```

```
if sum_val == K:  
    print(str(0))  
elif sum_val < K:  
    print(str((K - sum_val + N - 1) // N))  
else:
```

```
    upper = 10**9  
    lower = 0  
    while lower < upper:  
        mid = (lower + upper) // 2
```

```
        sum = 0  
        for x in v:  
            sum += max(1, x - mid)
```

```
        if sum > K:  
            lower = mid + 1
```

```
        else:  
            upper = mid
```

```
print(str(lower)) # o upper, hanno lo stesso valore
```