



INC REPORT PROBLEM M LC07

AOL Algorithm Design Analysis

Kelompok: C Level

Hanif Abdurrahman Ayyash - 2702314611

Pilar Nalendra Sarwanto - 2702362604

Zenia Nadia Rifaniputri - 2702343832

UNIVERSITAS BINA NUSANTARA

JAKARTA

2024

Problem M

ICPC Provincial

The University of INC (UOI) is participating in an ICPC Provincial Contest, a qualifier contest for the ICPC Regional Contest. UOI has $3N$ students (numbered from 1 to $3N$) who are eager to participate in the contest. There will be N teams, each consisting of exactly 3 students. Each student can only be assigned to only one team.

As the coach of UOI, you know that student i has a skill rating of A_i . You define the strength of a team as the median of the skill ratings of its members.

In order to increase the chance for all UOI teams to qualify for the ICPC Regional Contest, you want to arrange the teams so that the strength of the weakest team is maximized. Determine the maximum strength of the weakest team.

Input

The first line consists of an integer N ($1 \leq N \leq 100\,000$).

The second line consists of $3N$ integers A_i ($0 \leq A_i \leq 4000$).

Output

Output a single integer representing the maximum strength of the weakest team.

Sample Input #1

```
2
1500 1700 1800 2300 2500 2600
```

Sample Output #1

```
1800
```

Explanation for the sample input/output #1

Team 1 consists of students 1, 3, and 5, while team 2 consists of students 2, 4, and 6. The strength of team 1 and 2 are 1800 and 2300, respectively. Other arrangements exist, but none allow the weakest team to have a strength higher than 1800.

Sample Input #2

```
1
2800 2100 3000
```

Sample Output #2

2800

Explanation for the sample input/output #2

There is only one team with the strength of 2800.

Sample Input #3

3
 4000 0 4000 0 4000 0 4000 0 4000

Sample Output #3

0

Pada Problem M, kami menggunakan 2 algoritma:

1. Algoritma untuk mengurutkan array angka dari besar ke kecil, dan
2. Algoritma untuk mencari nilai median maksimum untuk tim terlemah

Untuk algoritma yang pertama, kami menggunakan merge sort. Merge sort dipilih karena metode ini memiliki time complexity terendah dibandingkan dengan metode sorting lainnya. Time complexity merge sort sebesar $n \log n$ atau $O(n \log n)$. Pertimbangan ini kami lakukan untuk mengantisipasi Time Limit pada judge.

Untuk algoritma yang kedua, kami menggunakan rumus:

$$\text{index} = N \times 2 - 1$$

Dimana:

Index : index array median tim terlemah

N : jumlah tim

Berikut adalah kronologi pembuatan rumus tersebut:

1. Andaikan jika, strength sebuah tim tidak didefinisikan berdasarkan median, namun berdasarkan skill rating terbesar dari tim tersebut. Jika demikian, setelah kita mengurutkan array skill rating dari terbesar ke terkecil, kita mau mengambil skill rating ke-N. Contoh:

N	= 1
Skill rating (setelah diurutkan)	= 3000, 2800, 2100

Karena hanya ada 1 tim, tim terlemah adalah satu-satunya tim yang ada tersebut. Berdasarkan definisi strength = skill rating terbesar, strength dari tim tersebut dan strength dari tim terlemah adalah skill rating ke-N, atau ke-1, atau 3000.

Contoh 2:

N = 2
Skill rating (setelah diurutkan) = 2600, 2500, 2300, 1800, 1700, 1500

Strength tim terlemah = skill rating ke-N, atau ke-2, atau 2500

2. Namun, berdasarkan pernyataan soal, strength didefinisikan oleh median dari skill rating tim terlemah. Jika demikian, kita hanya perlu memodifikasi rumusnya sedikit menjadi strength = skill rating ke-(N x 2). Mengapa demikian? Ini karena median selalu terletak di belakang skill rating terbesar. Jadi, untuk mencapai median dari tim terlemah, ibaratnya kita perlu melompati 2 skill rating (terbesar dan median) dari tiap-tiap tim. Kita tidak perlu melompati skill rating terendah tiap-tiap tim karena kita tidak perlu memaksimalkan skill rating terendah tiap-tiap tim, kita hanya perlu memaksimalkan median. Contoh:

N = 2
Skill rating (setelah diurutkan) = 2600, 2500, 2300, 1800, 1700, 1500

Strength tim terlemah = skill rating ke-(N x 2), atau ke-4, atau 1800

3. Lalu, karena penomoran index selalu dimulai dari 0, kita bisa memodifikasi rumusnya sedikit lagi menjadi strength = skill rating ke-(N x 2) - 1

Time complexity dari rumus ini N atau $O(n)$, dimana $n = N$ = jumlah tim.

Berikut adalah bukti kebenaran kode kami (dengan test case soal):

```
C:\Users\Hanif\Dropbox\PC\I  X  +  v
2
1500 1700 1800 2300 2500 2600
1800
-----
Process exited after 6.705 seconds with return value 0
Press any key to continue . . . |
```

```
C:\Users\Hanif\Dropbox\PC\I  X  +  v
1
2800 2100 3000
2800
-----
Process exited after 6.939 seconds with return value 0
Press any key to continue . . . |
```

```
C:\Users\Hanif\Dropbox\PC\I  X  +  v
3
4000 0 4000 0 4000 0 4000 0 4000
0
-----
Process exited after 4.048 seconds with return value 0
Press any key to continue . . . |
```