

TOKI Regular Open Contest #7 Editorial

(English version is available on page 5.)

Penulis soal

| | | |
|--------|---------------------------|----------|
| Div 2A | Prestasi Merah Putih | AMnu |
| Div 2B | Merah x Putih | AMnu |
| Div 1A | Di Antara Merah dan Putih | AMnu |
| Div 1B | Bangunan Merah Putih | AMnu |
| Div 1C | Kota Merah Putih | yogahmad |
| Div 1D | Pertarungan Merah Putih | AMnu |
| Div 1E | Rumah Sakit Merah Putih | yogahmad |

Div 2A. Prestasi Merah Putih

Solusinya adalah cukup melakukan looping pada semua nilai A_i yang ada, kemudian dibandingkan dengan nilai X .

Kompleksitas waktu: $O(N)$

Div 2B. Merah x Putih

Salah satu solusi paling sederhana adalah mencoba semua kemungkinan 8^5 nilai dari a , b , c , d , dan e , kemudian dikalikan masing-masing pasangannya dan dibandingkan dengan input P , Q , R , S , dan T .

Solusi lain yang lebih efisien adalah mencoba semua kemungkinan dari a yang membagi P . Jika kita mengetahui nilai a , maka nilai b bisa diperoleh dari P/a . Hal yang serupa untuk c , yang bisa diperoleh dari Q/b , dan seterusnya. Apabila semua pembagian bisa dilakukan tanpa sisa, maka kita sudah menemukan jawabannya.

Kompleksitas waktu: $O(\max(a)^5)$ atau $O(\max(a))$

Div 1A. Di Antara Merah Dan Putih

Apabila $\text{sum}(R) \leq \text{sum}(L) + 1$, tentu tidak ada jawaban karena di tidak ada bilangan bulat di antara $\text{sum}(L)$ dan $\text{sum}(R)$. Selain itu, maka dijamin terdapat sebuah solusi. Salah satu solusinya adalah, anggap $A = L$, kemudian cari salah satu elemen yang $A_i < R_i$, lalu tambahkan nilai A_i tersebut sebesar 1. Nilai A ini sudah bisa menjadi jawaban.

Kompleksitas waktu: $O(N)$

Div 1B. Bangunan Merah Putih

Jawaban dapat diperoleh dengan melakukan langkah-langkah berikut:

1. Hapus semua elemen ke- i yang memenuhi $H_{i-1} = H_i$ untuk $(2 \leq i \leq N)$.
2. Misalkan M adalah panjang array, setelah dilakukan semua langkah pertama.
3. Hapus semua elemen ke- i yang memenuhi $H_{i-1} < H_i < H_{i+1}$ atau $H_{i-1} > H_i > H_{i+1}$ untuk $(2 \leq i \leq M - 1)$.

Dengan melakukan langkah tersebut dapat dipastikan panjang array yang diperlukan sudah maksimal.

Kompleksitas waktu: $O(N)$

Div 1C. Kota Merah Putih

Soal ini dapat disederhanakan menjadi: Pilih sebuah edge untuk di-"potong" sedemikian sehingga diameter maksimum dari kedua tree yang diperoleh seminimum mungkin.

Perhatikan bahwa kita sebaiknya memotong tree di salah satu edge yang berada di diameter. Ini dapat dipastikan benar karena, jika kita memotong edge yang bukan berada di diameter, maka kita akan mendapatkan jawaban sama dengan diameter tree. Maka dari itu, kita bisa mencoba memotong satu per satu edge yang berada di diameter, kemudian cek diameter tree-nya pada setiap pemotongan. Hal ini bisa dilakukan dalam $O(N)$.

Dapat dibuktikan pula memotong edge di tengah-tengah diameter dapat memberikan jawaban yang optimal. Sehingga, kita hanya cukup memotong satu edge tersebut kemudian mengecek diameter dari kedua tree yang dihasilkan. Apabila diameter merupakan bilangan genap, kita bebas melakukan pemotongan pada salah satu dari kedua edge tengah tersebut dan dipastikan mendapat jawaban yang optimal.

Kompleksitas waktu: $O(N)$

Div 1D. Pertarungan Merah Putih

Misalkan $f(x, y)$ adalah sebuah fungsi yang bernilai 1 apabila petarung pertama dapat memenangkan pertarungan tersebut dengan sisa HP masing-masing x dan y .

Apabila $x \geq y$, tentu petarung pertama menang karena ia tinggal melakukan satu langkah penyerangan.

Apabila $2x \leq y$, maka petarung pertama sebaiknya bertahan, karena apabila ia menyerang, petarung kedua akan menyerang kembali dan membuat HP petarung pertama menjadi 0.

Apabila $x \leq B$, maka petarung pertama sebaiknya bertahan, karena apabila ia menyerang, petarung kedua akan bertahan dan HP-nya menjadi lebih besar atau sama dengan y (upaya penyerangan menjadi sia-sia).

Apabila petarung pertama harus bertahan (karena salah satu dari kedua kondisi di atas):

- Apabila $y \geq A$, maka petarung kedua akan menyerang dan membuat HP petarung pertama menjadi lebih kecil atau sama dengan x (upaya bertahan menjadi sia-sia). Dalam kasus ini, petarung pertama dipastikan tidak bisa menang
- Apabila $B \geq x + A$, maka petarung kedua akan bertahan pula, kemudian petarung pertama juga hanya bisa bertahan pada putaran selanjutnya (karena penyerangan akan sia-sia). Hal ini menyebabkan petarung pertama tidak bisa memenangkan pertarungan ini.
- Selain dari itu, petarung kedua menjadi sia-sia jika menyerang (karena $y < A$) dan bertahan (karena $B < x + A$) sehingga petarung pertama dapat dipastikan menang.

Apabila petarung pertama menyerang dan $y < A$, maka dapat dipastikan petarung pertama menang, karena setiap kali petarung kedua melakukan penyerangan, maka petarung pertama cukup bertahan dan membuat HP-nya kembali (penyerangan yang dilakukan petarung kedua sia-sia) dan setiap kali petarung kedua bertahan, maka petarung pertama cukup menyerang dan mengurangi HP lawan kembali (upaya bertahan yang dilakukan petarung kedua sia-sia).

Selain dari semua kondisi di atas, maka kita rekursi ke $f(2x - y, y - x)$ karena kedua petarung saling menyerang.

Sehingga apabila $f(x, y)$ bernilai 1, maka AMNU menang. Apabila salah satu dari $f(y, x+A)$ atau $f(y-x, x)$ bernilai 1 (dengan kemampuan masing-masing sudah ditukar), maka YOGA menang. Selain dari itu, pertarungan tidak bisa berhenti.

Kompleksitas waktu: $O(T \log \max(X, Y))$

Div 1E. Rumah Sakit Merah Putih

Soal ini dapat diselesaikan dengan Binary Search The Answer, Dijkstra, [SOS DP](#) dan [Hall's Marriage Theorem](#).

Kita dapat mengelompokkan setiap kota ke dalam salah satu dari $2^{\text{(Banyak Rumah Sakit)}}$ kelompok. Untuk suatu nilai X , suatu kota termasuk ke dalam kelompok MASK jika dan hanya jika BIT rumah sakit yang bernilai 1 memiliki jarak $\leq X$ dengan kota tersebut, dan BIT rumah sakit yang bernilai 0 memiliki jarak $> X$ dengan kota tersebut.

Misalkan fungsi $PASIEN(MASK)$ adalah jumlah semua pasien yang berada di kota yang termasuk kelompok MASK.

Misalkan fungsi $RUMAH_SAKIT(MASK)$ adalah jumlah semua kapasitas rumah sakit yang mana BIT rumah sakit pada MASK bernilai 1.

Untuk suatu jarak X , kita bisa memindahkan semua pasien ke rumah sakit jika dan hanya jika untuk setiap nilai MASK $\sum_{i \text{ subset dari MASK}} PASIEN(i) \leq RUMAH_SAKIT(MASK)$ (berdasarkan Hall's Marriage Theorem)

Kita bisa menghitung $\sum_{i \text{ subset dari MASK}} PASIEN(i)$ dengan menggunakan SOS DP.

Jawaban X yang paling minimal dapat dicari dengan menggunakan Binary Search.

Kompleksitas waktu: $O(2^K \times K \times \text{LOG}(10^9) + N + M \times K \times \text{LOG}(N))$

TOKI Regular Open Contest #7 Editorial

Problem authors

| | | |
|--------|--------------------------|-----------------|
| Div 2A | Red-White Accomplishment | AMnu |
| Div 2B | Red x White | AMnu |
| Div 1A | In Between Red and White | AMnu |
| Div 1B | Red and White Buildings | AMnu |
| Div 1C | Red and White Cities | yogahmad |
| Div 1D | Red White Duel | AMnu |
| Div 1E | Red White Hospital | yogahmad |

Div 2A. Red-White Accomplishment

The solution is to simply loop through the value of all A_i , then compare it with the value of X .

Time complexity: $O(N)$

Div 2B. Red x White

One of the ways is to find all 8^5 possible values for a , b , c , d , dan e , then multiply every of its pair and compare them with the input P , Q , R , S , dan T .

Another more efficient solution is to check all possible value of a that divides P . If we know the value of a , then b can be obtained from P/a . Similar for c , which can be obtained from Q/b , and so on. If all the division has no remainder, then we have found an answer.

Time complexity: $O(\max(a)^5)$ or $O(\max(a))$

Div 1A. In Between Red and White

If $\text{sum}(R) \leq \text{sum}(L) + 1$, then there is no answer since there is no integer between $\text{sum}(L)$ and $\text{sum}(R)$. Otherwise, there exist a solution. One way to construct it is to let $A = L$, then find one element that satisfies $A_i < R_i$, and add the value of the A_i by 1. This value of A is now an answer.

Time complexity: $O(N)$

Div 1B. Red and White Buildings

One way to obtain the answer is by the following steps:

1. Remove all of the i -th element that satisfies $H_{i-1} = H_i$ for all $(2 \leq i \leq N)$.
2. Let M be the length of the array after the first step.
3. Remove all of the i -th element that satisfies $H_{i-1} < H_i < H_{i+1}$ or $H_{i-1} > H_i > H_{i+1}$ for all $(2 \leq i \leq M - 1)$.

With the steps above, it can be guaranteed that the length of the required array is maximal.

Time complexity: $O(N)$

Div 1C. Red and White Cities

This problem can be simplified as: Choose an edge to be “cut” such that the maximum diameter of the two resulting trees is minimized.

Notice that we should cut an edge which lies on the diameter. This can be proven to be true because, if we cut an edge which does not lie on the diameter, then the answer will simply be the diameter. Therefore, we try to cut each of the edges from the diameter one by one, then check the trees' diameter after the cut. These can be done in $O(N)$.

It can be proven that cutting the middle edge of the diameter can give an optimal answer. Therefore, we simply cut one edge, and check the diameter of both the resulting trees. If the diameter is an even number, then we are free to cut any of two middle edges and guaranteed to give an optimal answer.

Time complexity: $O(N)$

Div 1D. Red White Duel

Let $f(x, y)$ be a function that returns 1 if the first warrior win the duel with remaining HP x and y respectively.

If $x \geq y$, then the first warrior will win as he simply do the attacking move once.

If $2x \leq y$, then the first warrior should defend, because if he attacks, the second warrior will attack back and depletes the first warrior's HP.

If $x \leq B$, then the first warrior should defend, because if he attacks, the second warrior will defend and his HP will become larger than or equal to y (the attacking attempt became in vain).

If the first warrior must defend (because of one of the two conditions above):

- If $y \geq A$, then the second warrior will attack and made the first warrior's HP become less than or equal to x (the defending attempt became in vain). In this case, the first warrior can not win the duel.
- If $B \geq x + A$, then the second warrior will defend, then the first warrior will defend too on the next turn (since attacking will be in vain). This will make the duel can not be won by the first warrior.
- Other than that, the first warrior can win the duel.

If the first warrior attack and $y < A$, then first warrior can win the duel, because every time the second warrior attack, the first warrior can simply defend and has its HP restored (the attacking done by the second warrior is in vain).

Other than all the conditions above, we simply recurse to $f(2x - y, y - x)$ since both of warriors are attacking each other.

Therefore if $f(x, y)$ equals to 1, then AMNU wins. If any of $f(y, x+A)$ or $f(y-x, x)$ is 1, then YOGA wins. Otherwise, the duel continues indefinitely.

Time complexity: $O(T \log \max(X, Y))$

Div 1E. Red White Hospital

This problem can be solved using Binary Search The Answer, Dijkstra, [SOS DP](#) and [Hall's Marriage Theorem](#).

We can group every city to one of the $2^{\text{(number of hospitals)}}$ group. For a value X , a city belongs to a group MASK if and only if the hospital BIT that has value of 1 has distance $\leq X$ to that city, and the hospital BIT that has value 0 has distance $> X$ to that city.

Let PATIENT(MASK) be a function that gives the sum of all patients belonging to the group MASK.

Let HOSPITAL(MASK) be a function that gives the sum of all hospital capacity for which the BIT of the MASK equals to 1.

For a distance X, we can move all the patients to the hospitals if and only if for every value of MASK, $\sum_{i \text{ subset dari MASK}} PASIEN(i) \leq RUMAH_SAKIT(MASK)$ (according to Hall's Marriage Theorem)

We can compute $\sum_{i \text{ subset dari MASK}} PASIEN(i)$ using SOS DP.

The answer of minimal X can be found using Binary Search.

Time complexity: $O(2^K \times K \times LOG(10^9) + N + M \times K \times LOG(N))$