

Codechef LEMUSIC

Little Elephant and Music

Prof. Edson Alves – UnB/FGA

The Little Elephant from the Zoo of Lviv likes listening to music.

There are N songs, numbered from 1 to N , in his MP3-player. The song i is described by a pair of integers B_i and L_i – the band (represented as integer) that performed that song and the length of that song in seconds. The Little Elephant is going to listen all the songs exactly once in some order.

The sweetness of the song is equal to the product of the length of that song and the number of different bands listened before (including the current playing song).

Help the Little Elephant to find the order that maximizes the total sweetness of all N songs. Print that sweetness.

Input

The first line of the input contains single integer T , denoting the number of test cases. Then T test cases follow. The first line of each test case contains single integer N , denoting the number of the songs. The next N lines describe the songs in the MP3-player. The i -th line contains two space-separated integers B_i and L_i .

Output

For each test, output the maximum total sweetness.

Constraints

- $1 \leq T \leq 5$
- $1 \leq N \leq 100000$ (10^5)
- $1 \leq B_i, L_i \leq 1000000000$ (10^9)

Exemplo de entradas e saídas

Sample Input

2

3

1 2

2 2

3 2

3

2 3

1 2

2 4

Sample Output

12

16

Solução com complexidade $O(N \log N)$

- Este problema pode ser resolvido por meio de um algoritmo guloso
- Considere uma ordenação arbitrária $\{m_1, m_2, \dots, m_N\}$ das músicas
- Considere que $j = i + 1$ e que a banda da música i já apareceu ao menos uma vez antes de i e que a banda j não apareceu antes de j
- Se as duas músicas trocarem de posição a contribuição da música j será a mesma, pois sua duração não muda e o número de bandas que já apareceram será o mesmo
- Contudo, a contribuição da música i aumenta, pois o número de bandas distintas é incrementado
- Assim, as músicas devem ser ordenadas de tal maneira que as B bandas distintas apareçam nas primeiras B posições ao menos uma vez

Solução com complexidade $O(N)$

- Por outro lado, seja $i < j$ índices de duas músicas da mesma banda, com $i \leq B$ e a duração de m_i seja maior do que a duração de m_j
- A troca de ambas músicas de posição melhora a resposta, uma vez que a maior duração será multiplicada por B e a menor duração será multiplicada pelo número de bandas distintas até i , que é um número menor ou igual a B
- Assim, a ordenação também deve colocar as músicas de menor duração de cada banda nas primeiras posições
- Por fim, $i < j$ são os índices de duas músicas consecutivas dentre as primeiras B posições, se a duração de m_i for maior do que m_j , a troca de posição das duas também melhora a resposta
- Portanto, as músicas devem ser ordenadas de tal modo que as músicas de menor duração de cada banda ocupe as B primeiras posições, em ordem de duração, e as demais ocupam as $N - B$ posições, em qualquer ordem

Solução AC com complexidade $O(N \log N)$

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4 using ll = long long;
5
6 struct Song { ll b, l; };
7
8 ll solve(vector<Song>& ms)
9 {
10     sort(ms.begin(), ms.end(), [](const Song& x, const Song& y) {
11         return x.l < y.l;
12     });
13
14     set<ll> bs;
15     vector<Song> pending;
16     ll ans = 0;
```

Solução AC com complexidade $O(N \log N)$

```
18     for (const auto& m : ms)
19     {
20         if (bs.count(m.b) == 0)
21         {
22             bs.insert(m.b);
23             ans += (m.l * bs.size());
24         } else
25             pending.push_back(m);
26     }
27
28     for (const auto& m : pending)
29         ans += (m.l * bs.size());
30
31     return ans;
32 }
33
34 int main()
35 {
36     ios::sync_with_stdio(false);
```


Solução AC com complexidade $O(N \log N)$

```
38  int T;  
39  cin >> T;  
40  
41  while (T--)  
42  {  
43      int N;  
44      cin >> N;  
45  
46      vector<Song> ms(N);  
47  
48      for (int i = 0; i < N; ++i)  
49          cin >> ms[i].b >> ms[i].l;  
50  
51      cout << solve(ms) << '\n';  
52  }  
53  
54  return 0;  
55 }
```