OJ 13095

Tobby and Query

Prof. Edson Alves – UnB/FGA

Problema

In his free time Tobby is always searching for interesting things. This time Tobby created the following problem: given a sequence of n integer numbers, Tobby would like to know how many different numbers are in the range $[l, r]$ $(r \geq l)$.

Entrada e saída

Input

The input has several test cases. The first line of each test case contains an integer n $(1 \leq n \leq 10^5)$, the size of the sequence of numbers. The next line contains n values a_i $(0 \leq a_i \leq 9)$, the numbers in the sequence. The next line contains an integer q $(1 \leq q \leq 10^4)$, the amount of queries. Then there are q lines, each line contains a query: two integers l and r $(1 \leq l, r \leq n)$.

Output

For each test case print q integers, representing the amount of different numbers in the range $[l, r]$ for each query in the input.

Exemplo de entradas e saídas

Sample Input

```
7
0 2 3 3 7 5 2
3
1 1
2 4
2 7
5
7 7 7 7 7
2
4 5
1 5
```

Sample Output

```
1
2
4
1
1
```

Solução $O(N + Q)$

- Uma forma de responder rapidamente (em $O(1)$) cada uma das consultas é calcular as somas dos prefixos p_d, onde d representa os 10 dígitos decimais (pois $0 \leq a_i \leq 9$)
- Estas somas podem ser computadas em $O(N)$
- Assim, a consulta para o intervalo $[L, R]$ pode ser respondida por meio da $RSQ(L, R)$ para cada um dos 10 vetores de prefixos:

$$q(L, R) = \sum_{d=0}^{9} \delta(p_d[R] - p_d[L-1]),$$

onde

$$\delta(x) = \begin{cases} 1, & \text{se } x > 0 \\ 0, & \text{caso contrário} \end{cases}$$

Solução $O(N + Q)$

```cpp
#include <bits/stdc++.h>

using namespace std;
using ii = pair<int, int>;

vector<int>
solve(int N, const vector<int>& xs, vector<ii>& qs)
{
    vector<vector<int>> ps(10, vector<int>(N + 1, 0));

    for (int i = 1; i <= N; ++i)
    {
        for (int d = 0; d <= 9; ++d)
            ps[d][i] += ps[d][i - 1];

        ps[xs[i]][i] += 1;
    }

    vector<int> ans;
```

Solução $O(N + Q)$

```
21      for (auto [L, R] : qs)
22      {
23          int res = 0;
24
25          for (int d = 0; d <= 9; ++d)
26              res += (ps[d][R] - ps[d][L - 1] > 0 ? 1 : 0);
27
28          ans.push_back(res);
29      }
30
31      return ans;
32  }
33
34  int main()
35  {
36      ios::sync_with_stdio(false);
37      int N;
38
39      while (cin >> N) {
40          vector<int> xs(N + 1);
```

Solução $O(N + Q)$

```
42          for (int i = 1; i <= N; ++i)
43              cin >> xs[i];
44
45          int Q;
46          cin >> Q;
47
48          vector<ii> qs(Q);
49
50          for (int i = 0; i < Q; ++i)
51              cin >> qs[i].first >> qs[i].second;
52
53          auto ans = solve(N, xs, qs);
54
55          for (auto x : ans)
56              cout << x << '\n';
57      }
58
59      return 0;
60  }
```