Universidade Federal de Pernambuco

las4s e pelados

Icaro Copo Papel Nunes, Joao Pou Grangeiro, Pedro Grisi

2025-12-26

Contest (1)

template.cpp

8 lines

```
#include <bits/stdc++.h>
using namespace std;

#define rep(i, a, b) for(int i = a; i < (b); ++i)
#define all(x) begin(x), end(x)
#define sz(x) (int)(x).size()
using ll = long long;
using pii = pair<int,int>;
```

.bashrc

2 lines

```
alias c='g++ -Wall -Wconversion -Wfatal-errors -g -std=c++17 \
  -fsanitize=undefined,address'
```

hash.sh

2 lines

```
# bash hash.sh file.cpp l1 l2
sed -n $2','$3' p' $1 | sed '/^#w/d' | cpp -dD -P -
     fpreprocessed | tr -d '[:space:]'| md5sum |cut -c-6
```

troubleshoot.txt

52 lines

```
Pre-submit:
Write a few simple test cases if sample is not enough.
Are time limits close? If so, generate max cases.
Is the memory usage fine?
Could anything overflow?
Make sure to submit the right file.

Wrong answer:
Print your solution! Print debug output, as well.
Are you clearing all data structures between test cases?
Can your algorithm handle the whole range of input?
Read the full problem statement again.
Do you handle all corner cases correctly?
Have you understood the problem correctly?
Any uninitialized variables?
Any overflows?
Confusing N and M, i and j, etc.?
Are you sure your algorithm works?
What special cases have you not thought of?
Are you sure the STL functions you use work as you think?
Add some assertions, maybe resubmit.
Create some testcases to run your algorithm on.
Go through the algorithm for a simple case.
Go through this list again.
Explain your algorithm to a teammate.
Ask the teammate to look at your code.
Go for a small walk, e.g. to the toilet.
Is your output format correct? (including whitespace)
Rewrite your solution from the start or let a teammate do it.

Runtime error:
Have you tested all corner cases locally?
Any uninitialized variables?
Are you reading or writing outside the range of any vector?
Any assertions that might fail?
Any possible division by 0? (mod 0 for example)
Any possible infinite recursion?
Invalidated pointers or iterators?
Are you using too much memory?
Debug with resubmits (e.g. remapped signals, see Various).

Time limit exceeded:
Do you have any possible infinite loops?
What is the complexity of your algorithm?
Are you copying a lot of unnecessary data? (References)
How big is the input and output? (consider scanf)
Avoid vector, map. (use arrays/unordered_map)
What do your teammates think about your algorithm?

Memory limit exceeded:
What is the max amount of memory your algorithm should need?
Are you clearing all data structures between test cases?
```

Data structures (2)

SegBeats.h

Description: In Segment Tree Beats, 'lazy' does NOT mean "updates still missing here". The node already reflects all previous updates. Instead, 'lazy' stores what must be propagated to the children before recursing. Always call 'apply(l,r,p)' before descending. This node layout supports range add, range chmin and range chmax operations. Beats conditions:
break: MIN x: mx1 <= x ; MAX x: mi1 >= x
tag: MIN x: x > mx2 ; MAX x: x < mi2
Time: amortized $\mathcal{O}\left(\log^2 N\right)$, without range add $\mathcal{O}\left(\log N\right)$

d41d8c, 47 lines

```
struct node{
  ll mx1, mx2, sum, lazy;
  ll mi1, mi2;
  int cMax, cMin, tam;
  node(int x=0) : mx1(x),mx2(-inf),mi1(x),mi2(inf),
                  cMax(1),cMin(1),tam(1),sum(x),lazy(0){}
  node(node a, node b){
    sum = a.sum+b.sum, tam = a.tam+b.tam;
    lazy = 0;
    mx1 = max(a.mx1, b.mx1);
    mx2 = max(a.mx2, b.mx2);
    if(a.mx1 != b.mx1) mx2 = max(mx2, min(a.mx1, b.mx1));
    cMax=(a.mx1==mx1 ? a.cMax:0)+(b.mx1==mx1 ? b.cMax:0);

    mi1 = min(a.mi1, b.mi1);
    mi2 = min(a.mi2, b.mi2);
    if(a.mi1 != b.mi1) mi2=min(mi2, max(a.mi1, b.mi1));
    cMin=(a.mi1==mi1 ? a.cMin:0)+(b.mi1==mi1 ? b.cMin:0);
  }
  void apply_sum(ll x){
    mx1 += x, mx2 += x,mi1 += x, mi2 += x;
    sum += tam*x, lazy += x;
  }
  void apply_min(ll x){
    if(x >= mx1) return;
    sum -= (mx1 - x)*cMax;
    if(mi1 == mx1) mi1 = x;
    if(mi2 == mx1) mi2 = x;
    mx1 = x;
  }
  void apply_max(ll x){
    if(x <= mi1) return;
    sum -= (mi1 - x)*cMin;
    if(mx1 == mi1) mx1 = x;
    if(mx2 == mi1) mx2 = x;
    mi1 = x;
  }
};
void apply(int l, int r, int p){
  for(int i=2*p+1; i<=2*p+2; i++){
    seg[i].apply_sum(st[p].lazy);
    seg[i].apply_min(st[p].mx1);
    seg[i].apply_max(st[p].mi1);
  }
  seg[p].lazy = 0;
}
```