American International University-Bangladesh

AIUB Eclipse

MD Siyam Talukder
Kazi Shoaib Ahmed Saad
Faysal Ahammed Chowdhury

Contents

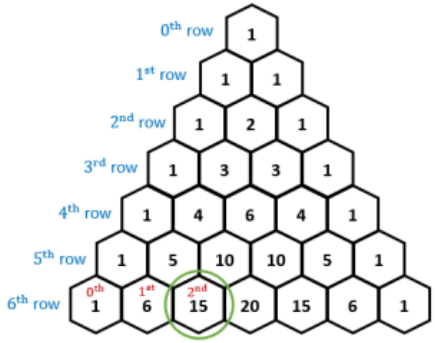
1 Number Theory

1.1 Code [45 lines]

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

const int N = 2e5 + 9, K = 20; // change
    here
int a[N];

void solve() {
  int n, q; cin >> n >> q;
  for (int i = 1; i <= n; i++) {
    cin >> a[i];
  }
  while (q--) {
    int l, r; cin >> l >> r;
    bool all_same = true;
    for (int i = l + 1; i <= r; i++) {
      all_same &= a[i] == a[i - 1];
    }
    if (all_same) {
      cout << 0 << ' ';
      continue;
    }
    int ans = 1;
    for (int x = 2; x <= 100; x++) {
      bool ok = true;
      for (int i = l; i <= r; i++) {
        ok &= (a[i] % x) == (a[l] % x);
      }
      if (ok) ans = x;
    }
    cout << ans << ' ';
  }
  cout << '\n';
}

int32_t main() {
  ios_base::sync_with_stdio(0);
  cin.tie(0);

  int t = 1; cin >> t;
  while (t--) {
    solve();
  }

  return 0;
}
```

1.2 hello [108 lines]

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

void solve() {
  int n; cin >> n;
  int a[n + 1], nxt[n + 1], prv[n + 1];
  for (int i = 1; i <= n; i++) {
    cin >> a[i];
  }
  memset(nxt, 0, sizeof nxt);
  memset(prv, 0, sizeof prv);

  set<int> odd, even;
  for (int i = 1; i <= n; i++) {
    if (i - 1 >= 1) prv[a[i]] = a[i - 1];
    if (i + 1 <= n) nxt[a[i]] = a[i + 1];
    if (i & 1) odd.insert(a[i]);
    else even.insert(a[i]);
  }

  int cur = a[1];
  for (int i = 1; i <= n; i++) {
    int who = 0;
    if (i & 1) {
      who = *odd.begin();
      odd.erase(odd.begin());
    }
    else {
      who = *even.begin();
      even.erase(even.begin());
    }

    if (who == cur) {
      cur = nxt[who];
      continue;
    }

    if (nxt[who] != 0) {
      nxt[prv[who]] = nxt[nxt[who]];
      prv[nxt[nxt[who]]] = prv[who];
      prv[who] = prv[cur];
      nxt[prv[cur]] = who;
      prv[cur] = nxt[who];
      nxt[nxt[who]] = cur;
    }
    else {
      if (prv[who] == nxt[cur]) {
        cur = nxt[who];
        continue;
      }

      int old_prv_who = prv[who];
      int old_nxt_who = nxt[who];
      int old_nxt_nxt_cur = nxt[nxt[cur]];
      int old_prv_cur = prv[cur];
      int old_nxt_cur = nxt[cur];

      nxt[prv[old_prv_who]] = old_nxt_who;

      prv[who] = prv[cur];
      nxt[prv[cur]] = who;
      nxt[who] = cur[nxt];

      prv[cur] = old_nxt_cur;
      nxt[cur] = old_prv_who;

      prv[old_nxt_cur] = who;
      nxt[old_nxt_cur] = cur;

      prv[old_prv_who] = cur;
      nxt[old_prv_who] = old_nxt_nxt_cur;
      prv[old_nxt_nxt_cur] = old_prv_who;
      if (nxt[old_nxt_nxt_cur] ==
          old_prv_who)
          nxt[old_nxt_nxt_cur] = 0;
    }

    cur = nxt[who];
  }


  int head = 0;
  for (int i = 1; i <= n; i++) {
    if (prv[i] == 0) {
      head = i;
      break;
    }
  }

  cur = head;
  while (cur != 0) {
    cout << cur << ' ';
    cur = nxt[cur];
  }
  cout << '\n';
}

int32_t main() {
  ios_base::sync_with_stdio(0);
  cin.tie(0);

  int t = 1; cin >> t;
  while (t--) {
    solve();
  }

  return 0;
}
```



1.3 Game Therory [0 lines]

1. Never use Graph
2. Always check time limit
3. Never use Graph
4. Always check time limit
5. Never use Graph as if we can never do this, Never use Graph as if we can never do this.
6. Always check time limit.

Here,
f[i] = how many pairs/k-tuple s.t. their gcd is i or it's multiple (count of pairs divisible by i)
g[i] = how many pairs/k-tuple such that their gcd is i
g[i] = f[i] - $\sum_{j|i} g[j]$

Sum of all pair gcd:
We know how many pairs have gcd = i for each i (1 to n). So the sum is

$$\sum_{i=1}^{n} g[i] \cdot i$$

Sum of all pair lcm (i = 1, j = 1):
We know

$$\text{lcm}(a, b) = \frac{a \cdot b}{\gcd(a, b)}$$

Now,
f[i] = All pair product sum of pairs whose gcd is i or multiple
g[i] = All pair product sum of pairs whose gcd is i

$$\text{Ans} = \sum_{i=1}^{n} \frac{g[i]}{i}$$

All pair product sum =

$$(a_1 + a_2 + \cdots + a_n) \cdot (a_1 + a_2 + \cdots + a_n)$$