奶龙大作战2题解

总览

- easy: A E F
- mid: B D
- hard: C

A

差分模板题

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

const int N = 5e6 + 10;
int a[N], d[N];

int main() {
    int n, p;
    std::cin >> n >> p;
    for (int i = 1; i <= n; ++i) {
        std::cin >> a[i];
    }

    for (int i = 1; i <= n; ++i) {
        d[i] = a[i] - a[i - 1];
    }

    while (p--) {
        int x, y, z;
        std::cin >> x >> y >> z;
        d[x] += z;
        d[y + 1] -= z;
    }

    int ans = INT_MAX;
    for (int i = 1; i <= n; ++i) {
        d[i] += d[i - 1];
        ans = std::min(ans, d[i]);
    }
    std::cout << ans << std::endl;
}
```

E

模拟

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

int main() {
    int t;
    std::cin >> t;
    while(t--) {
        std::string a, b;
        std::cin >> a >> b;
        std::swap(a[0], b[0]);
        std::cout << a << ' ' << b << std::endl;
    }
}
```

F

交互。询问前缀和，差分得到结果，对于a_1，询问区间$[2,3]$，区间$[1,3]$减去$[2,3]$即可

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

const int N = 1e3 + 5;
int n;
int a[N], pre[N];

int ask(int l, int r) {
    std::cout << "? " << l << ' ' << r << std::endl;
    std::cout.flush();
    int x;
    std::cin >> x;
    return x;
}

void ret() {
    std::cout << "! ";
    for (int i = 1; i <= n; ++i) {
        std::cout << a[i] << ' ';
    }
    std::cout << std::endl;
}

int main() {
    std::cin >> n;
    for (int i = n; i >= 2; --i) {
        pre[i] = ask(1, i);
    }

    for(int i = n; i >= 3; --i) {
        a[i] = pre[i] - pre[i - 1];
    }

    int x = ask(2, 3);
    a[2] = x - a[3];
    a[1] = pre[2] - a[2];

    ret();
```

```
}
```

B

dp。

$dp[0][i]$：表示走到第i行的线段的左端点的最少步数

$dp[1][i]$：表示走到第i行的线段的右端点的最少步数

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

const int N = 2e4 + 10;
int dp[2][N], l[N], r[N], len[N];

int dis(int a, int b) {
    return std::abs(a - b);
}

int main() {
    int n;
    std::cin >> n;
    for (int i = 1; i <= n; ++i) {
        std::cin >> l[i] >> r[i];
        len[i] = r[i] - l[i];
    }

    dp[0][1] = r[1] - 1 + len[1];
    dp[1][1] = r[1] - 1;
    for (int i = 2; i <= n; ++i) {
        dp[0][i] = std::min(dp[0][i - 1] + dis(r[i], l[i - 1]), dp[1][i - 1] + dis(r[i - 1], r[i])) + len[i] + 1;
        dp[1][i] = std::min(dp[0][i - 1] + dis(l[i - 1], l[i]), dp[1][i - 1] + dis(r[i - 1], l[i])) + len[i] + 1;
    }

    std::cout << std::min(dp[0][n] + dis(n, l[n]), dp[1][n] + dis(n, r[n])) << std::endl;
}
```

D

对顶堆模板题

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

int main() {
    int n, w;
    std::cin >> n >> w;
    std::priority_queue<int> a;
    std::priority_queue<int, std::vector<int>, std::greater<>> b;

    for(int i = 1; i <= n; ++i) {
        int x;
        std::cin >> x;
```

```
        if(b.empty() || b.top() < x) {
            b.push(x);
        } else {
            a.push(x);
        }
        int k = std::max(1, i * w / 100);
        while(b.size() > k) {
            a.push(b.top());
            b.pop();
        }
        while(b.size() < k) {
            b.push(a.top());
            a.pop();
        }
        std::cout << b.top() << ' ';
    }
}
```

C

最短路径模板题

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

using i64 = long long;
constexpr int INF = 1e9;

void solve() {
    int n, m, s, t;
    std::cin >> n >> m >> s >> t;
    std::vector<std::vector<std::pair<int, i64>>> adj(n);
    while(m--) {
        int u, v, w;
        std::cin >> u >> v >> w;
        u--;
        v--;
        adj[u].push_back({v, w});
        adj[v].push_back({u, w});
    }

    auto dijkstra = [&](int s, int t) {
        std::vector<int> dis(n, INF);
        std::priority_queue<std::pair<int, int>, std::vector<std::pair<int,
int>>, std::greater<>> pq;
        pq.emplace(0, s);

        while(!pq.empty()) {
            auto[d, u] = pq.top();
            pq.pop();

            if(dis[u] != INF) {
                continue;
            }
            dis[u] = d;

```

```
            for(const auto&[v, w] : adj[u]) {
                pq.emplace(d + w, v);
            }
        }
        return dis[t];
    };

    std::cout << dijkstra(s - 1, t - 1) << std::endl;
}

int main() {
    std::ios::sync_with_stdio(false);
    std::cin.tie(nullptr);
    std::cout.tie(nullptr);

    solve();
}
```