

```

struct line {
    long long m,b;
    line() {}
    line(long long __, long long ___) {
        m = __, b = ___;
    }
    long long get(long long x) {
        return m * x + b;
    }
};

struct cht {
    vector<line> trick;
    cht() { pos = 0;}
    int pos;
    bool irrelevant(line l1, line l2, line l3) {
        return (l3.b - l1.b) * (l1.m - l2.m) < (l2.b - l1.b) * (l1.m - l3.m);
    }
    void add(line l) {
        while(trick.size() >= 2
            and irrelevant(trick[trick.size() - 2], trick[trick.size() - 1], l))
            trick.pop_back();
        trick.push_back(l);
    }

    long long query(long long x) {
        if(pos >= (int)trick.size()) pos = trick.size() - 1;
        while(pos + 1 < (int)trick.size()
            and trick[pos + 1].get(x) < trick[pos].get(x))
            pos++;
        return trick[pos].get(x);
    }

    long long query2(long long x) {
        int a = -1;
        int b = trick.size() - 1;
        while(b - a > 1) {
            int mid = (a + b) / 2;
            if(trick[mid].get(x) > trick[mid + 1].get(x)) a = mid;
            else b = mid;
        }
        return trick[b].get(x);
    }
};

```