```cpp
/*
    | Convex Hull Trick |
    Desc: DP opt for problems involving linear functions.
    Source: KawakiMeido
    State: Untested lmao
*/

struct Line{
    int m,n;

    Line(int _m=0, int _n=0): m(_m), n(_n){};

    int operator()(const int& x) const { return m*x+n;};

    friend ld intersect (Line a, Line b) {
        return (ld)(b.n-a.n)/(ld)(a.m-b.m);
    };
};

struct LineContainer{
    deque<Line> dq;

    void add(Line line){
        while ((int)dq.size()>1 && intersect(dq[dq.size()-1],dq[dq.size()-2]) > intersect(dq[dq.size()-1],line)){
            dq.pop_back();
        }
        dq.push_back(line);
    }

    int getLine(int x){
        int ans = 0, l=1, r=dq.size()-1;
        while (l<=r){
            int mid = (l+r)/2;
            if (intersect(dq[mid],dq[mid-1])<=x){
                ans = mid;
                l = mid+1;
            }
            else r = mid-1;
        }
        return ans;
    }

    int getVal(int x){
        int idx = getLine(x);
        return dq[idx](x);
    }
} CHT;

```