```cpp
// 遅延(add) skew heap (最小値 heap)
// verified: http://judge.u-aizu.ac.jp/onlinejudge/review.jsp?rid=3355743
// new をやめると少し早くなる
template <typename T>
class LazySkewHeap
{
    struct Node
    {
        T val, add;
        Node *l, *r;

        Node(T v) : val(v), add(0), l(NULL), r(NULL) {}
        Node() {}
    };

    Node *root;

  private:
    void lazy(Node *a)
    {
        if (a->l)
            a->l->add += a->add;
        if (a->r)
            a->r->add += a->add;
        a->val += a->add;
        a->add = 0;
    }

    Node *meld(Node *a, Node *b)
    {
        if (!a)
            return b;
        if (!b)
            return a;
        // min Heap
        if (a->val + a->add > b->val + b->add)
            swap(a, b);
        lazy(a);
        a->r = meld(a->r, b);
        swap(a->l, a->r);
        return a;
    }

  public:
    LazySkewHeap() : root(NULL) {}

    void meld(LazySkewHeap *lsh)
    {
        root = meld(root, lsh->root);
    }

    void push(T v)
    {
        root = meld(root, new Node(v));
    }

    void add(T add)
    {
        assert(root != NULL);
        root->add += add;
    }

    T top()
    {
        assert(root != NULL);
        return root->val + root->add;
    }

    void pop()
    {
        assert(root != NULL);
        lazy(root);
        root = meld(root->l, root->r);
    }

    bool empty()
    {
        return !root;
    }
};

```