【题解】ZR3125 25省选10连测 day6-象形文字

主要部分见 ylx 的官方题解，本文只详细说明如何用小根堆维护差分值来求 dp。
dp 转移式：

$$F_{i,x} = \min(F_{i-1,x}, \max(F_{i-1,x-1}, b_i) + (a_i - b_i))$$

核心思路是 dp 值 $\le b_i$ 的前缀部分，dp 值只会从 $i-1$ 处继承，也就是说永远不会再变化；dp 值 $> b_i$ 的部分需要对 $F_{i-1,x-1} + a_i - b_i$ 取 min。
下面对着代码讲解。（代码抄的 cjr 的，拜谢 cjr）

```
int pos = 1; ll val = INF;
priority_queue<ll, vector<ll>, greater<ll> > que;
g[0] = 0;
for (int i = 1; i <= b_l; ++i) {
    while (pos <= i - 1 && val <= b[i].gain) { // @1
        g[pos++] = val;
        val += que.top();
        que.pop();
    }
    if (que.empty() && val <= b[i].gain) { // @2
        g[pos++] = val;
        val = INF;
    }
    else if (b[i].cost <= val) { // @3
        que.push(val - b[i].gain);
        val = b[i].cost;
    }
    else { // @4
        que.push(b[i].cost - b[i].gain);
    }
}
while (!que.empty()) { // @5
    g[pos++] = val;
    val += que.top();
    que.pop();
}
```

pos 表示第一个 $> b_i$ 的 dp 值的下标，val 表示 pos 处的 dp 值。小根堆中维护 $dp_{pos+1} - dp_{pos}, dp_{pos+2} - dp_{pos+1}, \ldots$ 这些差分值。
代码中 @1 处是将 $\le b_i$ 的 dp 值弹出。

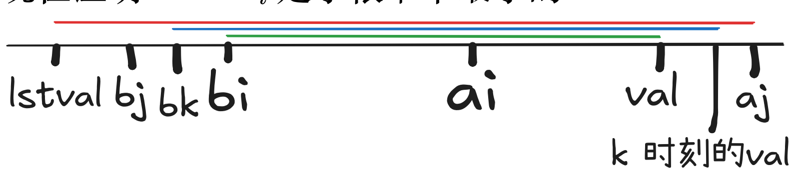
@2 处是在小根堆已经弹光了（即 $pos=i$）时，仍有 $val \le b_i$，则直接给 dp_{pos} 确定值，并将 val 设为 ∞（因为 dp_{pos+1} 不存在）。
@3 处是在 $a_i \le val$ 时，根据
$val = dp_{pos} = \max(dp_{pos-1}, b_i) + (a_i - b_i) = b_i + a_i - b_i = a_i$，应将 val 设为 a_i，并相应的更新小根堆。考虑应插入什么数。以 dp_{pos+1} 为例，

$$dp_{i,pos+1} = \min(dp_{i-1,pos+1}, \max(dp_{i-1,pos}, b_i) + (a_i - b_i))$$

$$= \min(dp_{i-1,pos+1}, a_i + (dp_{i-1,pos} - b_i))$$

$$= \min(dp_{i-1,pos+1}, dp_{i,pos} + (\text{更新前的}val - b_i))$$

因此插入的差分值应为 $\text{更新前的}val - b_i$。
现在还需要保证这一值是小根堆里最小的，才能使得它位于序列最前面成为 $dp_{pos+1} - dp_{pos}$，才能保证正确性。证明见下文。
@4 处也是根据 dp 式子而更新的小根堆。显然根据 dp 式子可以得出插入后差分值的序列为原差分序列加入 $a_i - b_i$，且仍然是有序的（懒得写证明了，很简单）。于是可以直接插入到小小根堆里。
@5 处是处理小根堆里剩余元素，也就是 dp_{pos} 到 dp_n 这些尚未赋值过的地方。
现在证明 $val - b_i$ 是小根堆中最小的。



如图，在数轴上，val 为当前的 val，$val - b_i$ 为图中绿线的长度。
设 $lstval$ 为上一次在 @1 处弹出时 val 的值，弹出后 val 变为了某一个 a_j，此时（弹出前）小根堆的 `top()` 为图中红线的长度，显然它 $\ge val - b_i$。显然此时小根堆里剩余的元素均 $\ge$ `top()`，进而 $\ge val - b_i$。于是我们只需要证明在此之后小根堆里新插入的元素 $\ge val - b_i$ 即可。
然后，程序进入了若干次 @3 处，容易发现在这个过程中 val 只会逐渐变小，最终变为现在的 val。设 a_k, b_k 为这其中的某个时刻的 a, b 值，于是 $\text{这时的}val$ 必定 $\ge \text{现在的}val$。由于是按 b 从小到大排序的，显然 $b_k \le b_i$。从而，这些时刻插入的值 $\text{这时的}val - b_k$（图中蓝线）必定 $\ge \text{现在的}val - b_i$。
当程序进入 @4 时，必然有 $a \ge \text{现在的}val, b \le b_i$。于是这些时刻插入的值 $\ge val - b_i$

。

综上所述，$val - b_i$ 是小根堆中最小的。