

知识精炼（三）



主讲人：邓哲也



BZOJ 2678 Bookshelf

现在有 N 本书需要按顺序放进书架，第 i 本书宽 $w[i]$ ，高 $h[i]$ 。

每个书架最多可以存放宽度和为 L 的书，这个书架的高度应该是里面的书中最高的一本书的高度。

求每个书架高度之和的最小值。

$N \leq 200000, 1 \leq L \leq 10^9$

BZOJ 2678 Bookshelf

样例：（答案：21）

N L

$h[i]$ $w[i]$

5 10

5 7

9 2

8 5

13 2

3 8

BZOJ 2678 Bookshelf

由于书架的个数没有限制，我们只需要记录一维状态即可。

用 $f[i]$ 表示把前 i 本书装进书架得到的书架高度之和的最小值。

枚举从第 i 本书开始到哪本书为止放进一个新的书架。

比如把 $[j, i]$ 这些书放入一个新书架。

产生的代价是 $f[j - 1] + \max\{h[j], h[j + 1], \dots, h[i]\}$

前提是 $w[j] + w[j + 1] + \dots + w[i] \leq L$

BZOJ 2678 Bookshelf

用 $\text{sum}[i]$ 表示前 i 本书的宽度之和。

$$f[i] = \min\{f[j-1] + \max\{h[j], h[j+1], \dots, h[i]\} \mid 0 \leq j < i, \text{sum}[i] - \text{sum}[j-1] \leq L\}$$

直接暴力枚举 j （倒着枚举，边枚举边计算 h 的最大值），
时间复杂度是 $O(N^2)$

BZOJ 2678 Bookshelf

$$f[i] = \min\{f[j-1] + \max\{h[j], h[j+1], \dots, h[i]\} \mid 0 \leq j < i, \text{sum}[i] - \text{sum}[j-1] \leq L\}$$

我们先来看看 $\max\{h[j], h[j+1], \dots, h[i]\}$ 随着 i 的右移会发生什么。

记录 $a[j] = \max\{h[j], h[j+1], \dots, h[i]\}$

那么随着 i 的右移, $a[j]$ 一定会单调递增。

BZOJ 2678 Bookshelf

记录 $a[j] = \max\{h[j], h[j+1], \dots, h[i]\}$

假设此时 $i = i + 1$

那么我们找到小于 $i + 1$ 的第一个大于 $h[i + 1]$ 的位置 k

($h[k] > h[i + 1]$)

对于 $[k + 1, i + 1]$ 这个区间的 a 值全都变成了 $h[i+1]$

小于等于 k 的部分，因为 $h[k] > h[i + 1]$ ，因此 a 值不会发生变化。

BZOJ 2678 Bookshelf

$$f[i] = \min\{f[j-1] + a[j] \mid 0 \leq j < i, \text{sum}[i] - \text{sum}[j-1] \leq L\}$$

由于 $0 \leq j < i$, $\text{sum}[i] - \text{sum}[j-1] \leq L$ 可以得到 j 的可行值是一个区间。

因此我们只要在下标 j 的位置维护 $f[j-1] + a[j]$

然后更新 $f[i]$ 的时候，在一个区间上查询最小值就可以了。

我们可以用线段树来维护区间更新和区间查询操作。

BZOJ 2678 Bookshelf

问题变成了对每个数，找到它左边第一个大于它的数。

维护一个单调下降的队列就可以了。

每次进来一个 $h[i + 1]$ 如果大于队尾元素，那就说明从队尾这个下标到 $i + 1$ 这个位置上的 a 都应该变成 $h[i + 1]$ ，删除队尾元素，继续检查。

另外还需要检查队头元素到 $i + 1$ 的宽度之和是否小于等于 L ，不满足的话需要删除。

时间复杂度 $O(N \log N)$

BZOJ 2678 Bookshelf

```
#define ls (x << 1)
#define rs (x << 1 | 1)
void upd(int x) {
    seg[x] = min(seg[ls], seg[rs]);
}
void down(int x) {
    if(add[x]) {
        seg[ls] += add[x];
        seg[rs] += add[x];
        add[ls] += add[x];
        add[rs] += add[x];
        add[x] = 0;
    }
}
```

BZOJ 2678 Bookshelf

```
void modify(int ql, int qr, ll v, int l, int r, int x) {  
    if(ql <= l && r <= qr) {  
        seg[x] += v;  
        add[x] += v;  
        return;  
    }  
    int mid = (l + r) >> 1;  
    down(x);  
    if(ql <= mid) modify(ql, qr, v, l, mid, ls);  
    if(mid < qr) modify(ql, qr, v, mid + 1, r, rs);  
    upd(x);  
}
```

BZOJ 2678 Bookshelf

```
11 ask(int ql, int qr, int l, int r, int x) {  
    if(ql <= l && r <= qr) return seg[x];  
    int mid = (l + r) >> 1;  
    ll ret = inf;  
    down(x);  
    if(ql <= mid) ret = min(ret, ask(ql, qr, l, mid, ls));  
    if(mid < qr) ret = min(ret, ask(ql, qr, mid + 1, r, rs));  
    return ret;  
}
```

BZOJ 2678 Bookshelf

```
int main() {
    scanf("%d%d", &n, &L);
    for(int i = 1; i <= n; i++) {
        scanf("%d%d", &h[i], &w[i]);
        sum[i] = sum[i - 1] + w[i];
    }
    memset(seg, 63, sizeof(seg));
    s = 1, e = 0;
    int l = 1;
```

BZOJ 2678 Bookshelf

```
for(int i = 1;i <= n;i ++){
    q[++ e] = i;
    v[e] = h[i];
    while(s < e && v[e - 1] < v[e]){
        modify(q[e - 1],q[e] - 1,v[e] - v[e - 1],1,n,1);
        v[e - 1] = v[e];
        -- e;
    }
    modify(i,i,-inf + h[i] + f[i - 1],1,n,1);
    while(1 <= i && sum[i] - sum[1 - 1] > L) 1 ++;
    f[i] = ask(1,i,1,n,1);
}
cout << f[n] << endl;
return 0;
}
```

下节课再见