前缀数组优化

主讲人：邓哲也

BZOJ 1705 Telephone Wire

新的电话线架设在已有的 N(2≤N≤100000) 根电话线杆上，第 i 根电话线杆的高度为 h[i](1≤h[i]≤100)。如果两根电话线杆的高度不同，那么就需要支付 C * 电话线杆高度差(1≤C≤100)的费用。你不能移动电话线杆，只能按照原有的顺序在相邻的线杆间架设电话线。

当然你也可以加高某些电话线杆，加高 X 米需要付出 X^2 的费用。

请问最少需要花多少钱建设新的电话线。

样例：

N=5,C=2

2 3 5 1 4

答案：15 （改造为 3 3 5 3 4）

BZOJ 1705 Telephone Wire

基本思路：

观察发现 h[i] 不大，可以作为状态。

用 f[i][j] 表示第 i 根电线杆的长度为 j 时的最小代价。

状态转移：

$f[i][j] = \min\{f[i-1][k] + c * |j - k| + (j - h[i])^2\}$

BZOJ 1705 Telephone Wire

$f[i][j] = \min\{f[i-1][k] + c * |j - k| + (j - h[i])^2\}$

这样的时间复杂度是 $O(NK^2)$

状态有 NK 个，瓶颈在于计算每个状态需要枚举前一个电线杆的可能的 K 种高度。

$f[i][j] = \min\{f[i-1][k] + c * |j - k|\} + (j - h[i])^2$

但是前一部分还是有 j 在，如果我们能把 j 和 k 分离，我们就可以方便的优化。

BZOJ 1705 Telephone Wire

$f[i][j] = \min\{f[i-1][k] + c * |j - k|\} + (j - h[i])^2$

我们可以把绝对值拆掉，也就是分类讨论：

如果 $j > k$，那么 $|j - k| = j - k$

$f[i][j] = \min\{f[i-1][k] - ck\} + cj + (j - h[i])^2$

如果 $j < k$，那么 $|j - k| = k - j$

$f[i][j] = \min\{f[i-1][k] + ck\} - cj + (j - h[i])^2$

BZOJ 1705 Telephone Wire

我们用 p[i][j] 维护 min{f[i][k] - ck | k≤j}

用 q[i][j] 维护 min{f[i][k] + ck | k≥j}

如果 j > k，f[i][j] = p[i-1][j] + cj + (j - h[i])²

如果 j < k，f[i][j] = q[i-1][j] - cj + (j - h[i])²

这样每次转移都是 O(1) 的。

每一轮循环都更新一次 p，q 数组即可。

时间复杂度就是 O(NK)

第一维也是没用的，我们可以用滚动数组优化掉。

BZOJ 1705 Telephone Wire

```
int main(){
	scanf(“%d%d”, &n, &c);
	for(int i = 1;i <= n;i ++) scanf(“%d”, &h[i]);
	memset(dp, 63, sizeof(dp));
	for(int i = h[1];i <= 100;i ++) dp[i] = sqr(i - h[1]);
	for(int i = 2;i <= n;i ++){
		f[i] = 0x3f3f3f3f;
		for(int j = 100;j >= 1;j --)
			f[j] = min(dp[j] + j * c, f[j + 1]);
		g[0] = 0x3f3f3f3f;
		for(int j = 1;j <= 100;j ++)
			g[j] = min(dp[j] - j * c, g[j - 1]);
```

BZOJ 1705 Telephone Wire

```
            for(int j = 0;j <= 100;j ++)
                dp[j] = 0x3f3f3f3f;
            for(int j = h[i];j <= 100;j ++)
                f[j] = sqr(j - h[i]) + min(g[j] + j * c, f[j] - j
* c);
        }
        int ans = 0x3f3f3f3f;
        for(int i = 1;i <= 100;i ++)
            ans = min(ans, dp[i]);
        printf("%d\n", ans);
        return 0;
}
```

下节课再见