

J. LEVELING GROUND

PROBLEM DESCRIPTION

題目給一個折線圖，現在要剷平土地，剷除花費根據地形有不同的成本；

題目問若要將一連續且長度為 M 的區塊剷平，會需要花多少成本

SOLUTION TECHNIQUES

線段樹、Sliding Window

SOLUTION SKETCHES

還滿直覺的，我們移動一個長度為 M 的 Sliding Window，並用一顆線段樹紀錄區間最低高度，如此我們每移動一單位僅需 $O(\log N)$ 的時間。

細節讓這題不是很好寫。

TIME COMPLEXITY

每筆測資 $O(N \log N)$ ， N 為土地總長度。

SOLUTION PROGRAM FOR REFERENCE

```
#include <iostream>
#include <cstdio>
#include <cstring>
#include <algorithm>

using namespace std;
typedef long long ll;

const int N = 1000005;
const int M = -N;
const int inf = 1e9;

char a[N];
int h[N];
int s[N << 2];

void build(int i, int l, int r)
{
    if (r == l) { s[i] = h[l]; return ; }
    int m = (l + r) >> 1;
    build(i << 1, l, m);
    build(i<<1|1, m + 1, r);
    s[i] = min(s[i << 1], s[i<<1|1]);
}

int que(int i, int l, int r, int ql, int qr)
{
    if (ql > r || qr < l) return inf;
```

```

        if (ql <= l && r <= qr) return s[i];
        int m = (l + r) >> 1;
        return min( que(i << 1, l, m, ql, qr), que(i<<1|1, m + 1, r,
ql, qr) );
    }

```

```

int add(int pos)
{
    return (h[pos] - M) * 2 + (a[pos] != '_');
}

```

```

int main()
{
    int i, ch, tt, n, m;
    ll tmp, ans;
    scanf("%d", &tt);
    for (int cc = 1; cc <= tt; cc++)
    {
        ans = 1e18;
        ch = 0;
        scanf("%d%d ", &n, &m);
        gets(a);
        h[0] = 0;
        for (i = 0; a[i]; i++)
        {
            h[i] = ch;
            if (a[i] == '\\') h[i]--;
            if (a[i] == '/')
                ch++;
            else if (a[i] == '\\')

```

```

        ch--;
    }
    //for (i = 0; i < n; i++) printf("%d ", h[i]);
puts("");
    build(1, 0, n - 1);
    tmp = 0;
    for (i = 0; i < m; i++)
        tmp += add(i);
    // tmp -= (ll)m * (que(1, 0, n - 1, 0, m - 1) - M) * 2;
    ans = tmp - (ll)m * (que(1, 0, n - 1, 0, m - 1) - M) *
2;
    for (i = 1; i <= n - m; i++)
    {
        tmp += add(i + m - 1);
        tmp -= add(i - 1);
        ans = min(ans, tmp - (ll)m * (que(1, 0, n - 1, i,
i + m - 1) - M) * 2);
    }
    printf("Case #%d: %.1f\n", cc, (double)ans * 0.5);
}
return 0;
}

```