B – Moogle 解题报告

[Source]

http://poj.org/problem?id=3638

[Description]

用地图软件记录一条大街上（可看做直线）的 h 栋房子的信息，但由于存储空间有限，所以只能存储 c 栋房子的位置（c ≤ h），剩下的房子位置采用线性插值的方法估算，求最后的平均误差。

关于线性插值：看了好久才看懂了是怎样估计的，主要还是从 x[i]+(x[j]-x[i])*(k-i)/(j-i) 中看出来的。用 sample 2 来说明：

INPUT

```
10 4
0 10 19 30 40 90 140 190 202 210
```

OUTPUT

```
0.3000
```

最小的误差方案是选择存储 0、40、190、210 四个位置处的房子信息。
通过线性插值估算后得到的全部房子位置为：

```
0 10 20 30 40 90 140 190 200 210
```

即存在 fabs(20-19) 和 fabs(200-202) 的误差，所以最后输出的结果为：

```
(fabs(20-19)+fabs(200-202))/4
```

[Solution]

可以用一个数组 error[][]记录线性插值的误差：
error[i][j]表示两端存储第 i 个位置和第 j 个位置的房子位置后线性插值的误差。
在用 dp[i][j]表示处理完第 i 个房子已经记录了 j 个房子位置的最小总误差。
状态方程：dp[i][j] = min{dp[i][j], dp[k][j-1]+error[i][k]} (j-2<=k<=i-1)

[Code]

（代码没什么优化，就是赤裸裸的 3 重循环了。另外 dp 可能有很多不必要的初始化，凑合着看吧）

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>
using namespace std;
```

```
#define N 210
#define inf 1000000000
#define see(x) cout<<#x<<":"<<x<<endl;
int a[N];
double error[N][N],dp[N][N];
void init(int n){
	int i, j, k;
	double inter, s;
	for(i=0;i<n;i++){
		for(j=i+1;j<n;j++){
			inter = double(a[j]-a[i])/(j-i);
			s = 0.0;
			for(k=i+1;k<j;k++){
				s += fabs(a[i]+inter*(k-i)-a[k]);
			}
			error[i][j] = error[j][i] = s;
		}
	}
}

int main(){
	int t, h, c;
	int i, j, k, l, m;
	scanf("%d",&t);
	while(t--){
		scanf("%d%d",&h,&c);
		for(i=0;i<h;i++){
			scanf("%d",&a[i]);
		}
		init(h);
		for(i=0;i<h;i++){
			for(j=1;j<=c;j++){
				dp[i][j] = inf;
			}
		}
		dp[0][1] = 0;
		for(i=1;i<h;i++){
			for(j=2;j<=c&&j-1<=i;j++){
				dp[i][j] = min(dp[i][j],dp[i-1][j-1]);
				for(k=j-2;k<i-1;k++){
					dp[i][j] =
min(dp[i][j],dp[k][j-1]+error[i][k]);
				}
			}
```

```
                }
                printf("%.4f\n",dp[h-1][c]/h);
        }
}
```