

# Grazing on the Run

## 一. 考察内容:

动态规划 区间DP

## 二. 题目分析:

[题目大意]

给出一些点的位置，和起点，要求每个点至少到达一次，每个点有一个不新鲜值，第一次到达该点时不新鲜值停止计算，求出总体最少的不新鲜值之和。

[写题思路]

考虑区间DP，设 $f[i][j]$ 为区间 $[i,j]$ 所有点都访问之后获得的最小总不新鲜度，特殊的， $f[i][j][0]$ 表示最后停在区间左端的情况， $f[i][j][1]$ 表示最后停在区间右端的情况。

转移，很显然， $f[i][j][0]$ 一定是由 $f[i+1][j]$ 转移过来的， $f[i][j][1]$ 一定是由 $f[i][j-1]$ 转移过来的，则每种状态由他之前的状态从两端转移而来即可。

## 三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATED
/*****
*创建时间: 2018 09 16
*文件类型: 源代码文件
*题目来源: BZOJ
*当前状态: 已通过
*备忘录: 动态规划 区间DP
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
#include <bitset>
// #include <sys/wait.h>
// #include <sys/types.h>
// #include <unistd.h>
using namespace std;
#define MAXN 1001
int n,a[MAXN],f[MAXN][MAXN][2];
/* Variable explain:

*/
void read()
{
    int ls1;
    scanf("%d%d",&n,&ls1);
    for(int i=1;i<=n;++i)for(int j=1;j<=n;++j)f[i][j][0]=f[i][j][1]=2e6;
    for(int i=1;i<=n;++i)scanf("%d",&a[i]);
    sort(a+1,a+1+n);
    for(int i=1;i<=n;++i)f[i][i][0]=f[i][i][1]=abs(ls1-a[i])*n;
    return;
}
int main()
{
    // freopen(".in","r",stdin);
    // freopen(".out","w",stdout);
    read();
    for(int i=2;i<=n;++i)//len
    {
        for(int j=1;j<=n-i+1;++j)//start
```

```
    {
        int k=i+j-1;
        f[j][k][0]=min(f[j+1][k][0]+(a[j+1]-a[j])*(n-i+1),f[j+1][k][1]+(a[k]-
a[j])*(n-i+1));
        f[j][k][1]=min(f[j][k-1][0]+(a[k]-a[j])*(n-i+1),f[j][k-1][1]+(a[k]-
a[k-1])*(n-i+1));
    }
    printf("%d\n",min(f[1][n][0],f[1][n][1]));
    return 0;
}
```

[<题目跳转>](#) [<查看代码>](#)