气象牛

一. 考察内容：

动态规划

二. 题目分析：

[题目大意]

给出一个长度为n的序列，让你从中挑出最少的数，使得挑出的序列与原序列的差异度不超过k，输出挑出的数的个数和最小的差异度。

[写题思路]

这是一道很经典的动态规划题，之所以经典，因为他可以很清晰的告诉我们如何去思考状态的设定，设定状态不能死记硬背，需要通过分析题目得出该题需要维护哪些**关键信息**才可以得出答案，以及该题需要维护哪些**周边信息**才可以**简单的、正确的**转移关键信息。

考虑设状态f[i][j][k]为前i个元素，选出j个元素，最后一个倒数第二个选的是k时的最少最少花费，转移时枚举f[k][j-1][l]来转移，即$f_{ijk}=min(f_{ij*})+\sum_{l=k+1}^{i-1}|2*a_l-a_i-a_k|$。

我们发现我们这样设计状态，最后的答案并不是真正的最小花费，还需要加上i点之后点的权值，所以我们考虑再枚举每个状态结尾所需的权值（当然这个是可以优化枚举的），最后统计一下满足要求的最小值即可。

三. 代码实现：

```
#define _CRT_SECURE_NO_DEPRECATE
/*************************
*创建时间：2018 09 04
*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 二分答案
*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;
int n,m,a[101],f[101][101][101],minf[101][101];
/* Variable explain:

*/
void read()
{
    scanf("%d%d",&n,&m);
    for(int i=1;i<=n;++i)scanf("%d",&a[i]);
    return;
}
void dp()
{
    //f[i][j][k]:前i个节点，选择j个（i点也选择），前一个是第k的个最小误差
    for(int i=1;i<=n;++i)for(int j=1;j<=n;++j)for(int k=1;k<=n;++k)f[i][j][k]=1e9;
```

```
    for(int i=1;i<=n;++i)
    {
        for(int j=1;j<=i;++j)
        {
            minf[i][j]=1e9;
            if(j==1)//之前没有选出的样本
            {
                f[i][j][i]=0;
                for(int k=1;k<i;++k)f[i][j][i]+=abs(a[i]-a[k])<<1;
                minf[i][j]=min(minf[i][j],f[i][j][i]);
                // printf("f[%d][%d]=%d\n",i,j,minf[i][j]);
                continue;
            }
            for(int k=max(1,j-1);k<=i;++k)
            {
                // for(int l=max(1,j-2);l<k;++l)f[i][j][k]=min(f[i][j][k],f[k][j-1]
[l]);//找到最小状态
                f[i][j][k]=minf[k][j-1];
                for(int l=k+1;l<i;++l)
                {
                    f[i][j][k]+=abs(2*a[l]-(a[i]+a[k]));
                }
                minf[i][j]=min(minf[i][j],f[i][j][k]);
            }
            // printf("f[%d][%d]=%d\n",i,j,minf[i][j]);
        }
    }
    int ans=1e9,nowans,minnode=1000;
    for(int i=1;i<=n;++i)
    {
        for(int j=1;j<=min(n,minnode);++j)
        {
            nowans=0;
            for(int k=i+1;k<=n;++k)
            {
                nowans+=2*abs(a[k]-a[i]);
            }
            // printf("%d %d %d %d\n",i,j,nowans,minf[i][j]+nowans);
            if(j<minnode&&minf[i][j]+nowans<=m)minnode=j,ans=nowans+minf[i][j];
            if(j==minnode)ans=min(ans,nowans+minf[i][j]);
        }
    }
    printf("%d %d\n",minnode,ans);
}
int main()
{
    // freopen(".in","r",stdin);
    // freopen(".out","w",stdout);
    read();
    dp();
    return 0;
}
```

<题目跳转> <查看代码>