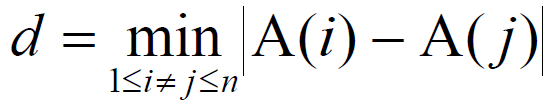
Home Work 4

令A[1..n]是一个由n个数所组成的数组，定义



为数组A的差值，其中|a|表示a的绝对值。设计一个求数组差值的算法（用伪码描述）并分析算法的时间复杂性。

## **解决**:

数组A有n个元素如下：[A0,A1,A2,A3,...,An]，即创建新的数组B，B中有n-1个元素，分别是：

B0=A0-A1；

B1=A1-A2;

B2=A2-A3;

······

Bn-1=An-1-An-2;

假设数组A的差值为|Ai-Aj|，那么，对应的，|Ai-Aj|=|Bi + Bi+1 +...+ Bj-1|，显然，求数组A的差值问题就被转换成了求数组B的最大连续子序列和的绝对值的问题。

而通过动态规划，实现算法复杂度为O(n)。

#include <stdio.h>

int main()

{

int A[Asize] =A1,A2,···,An;

int B[Bsize] ;//创建数组

int maxhere,maxsum;

for(int i =0; i < Asize-1; i++)

{

B[i]= A[i] - A[i+1];

} //向数组B中添加元素Ai-Ai+1

//动态规划实现过程

maxhere=maxsum=B[0];

for(int i = 1; i < Bsize; i++)

{

if((maxhere+B[i]) < B[i])

{

maxhere = B[i];

}

/\*如果前面位置到A[i]的最大子序列和

绝对值反而比|A[i]|要小，则以当前位置i结尾的最大子序列和为A[i]\*/

else

{

maxhere += B[i];

}

/\*如果前面位置到A[i]的最大子序列和

绝对值要比之前位置的更大，则取最大子序列和绝对值为两者之和\*/

if(maxhere > maxsum)

{

maxsum = maxhere;

}//更新最大子序列和的绝对值

}

}

创建数组B的算法复杂度为O(n-1)，实现最大子序列和绝对值的算法复杂度为O(n-1)，则总体算法复杂度为

O(n-1)+O(n-1)=O(n).