

20180329_阿里一面笔试题目

题目如下：

给定两个数组 $a(1), a(2), \dots, a(m)$, $b(1), b(2), \dots, b(n)$. 把数组 a 插入到 b 中，注意合并后的数组， a 的相对次序没有变，比如， $a(1)$ 还是在 $a(2)$ 的前面.

给一个例子：

$[a_1, a_2, a_3, a_4, a_5]$

$[b_1, b_2, b_3, b_4, b_5]$

合并后，可能是

$[a_1, a_2, b_1, b_2, a_3, b_3, a_4, b_4, a_5, b_5]$ # 1

$[b_1, b_2, a_1, b_3, a_2, a_3, a_4, b_4, b_5, a_5]$ # 2

比如针对第二种情况：相邻乘积和是：

$b_1 * b_2 + a_1 * b_3 + a_2 * a_3 + a_4 * b_4 + b_5 * a_5$

要求所有可能中的最小值：

针对这种情况，第一种想法就是全遍历，用dfs，如下。

另一种是用dp，

$f[n][m]$ 表示 A 中的 n 个数， B 中的 m 个数，求得的组合数组后，两两相乘的累积和。

$f[n][m] = \min\{f[n][m-2] + B[m-1]*B[m], f[n-1][m-1] + A[n]*B[n], f[n-2][m] + A[n-1]*A[n-2]\}$

下面我漏判了一个， $n+m$ 必须是偶数，才能配对。

示例代码如下：

```
#include <iostream>
#include <bits/stdc++.h>

using namespace std;

int min_value = INT_MAX;

void dfs( int A[], int B[], int startA, int lenA, int startB, int lenB,
int & sum ){
    if( lenA == 0 && lenB == 0 ){
        return;
    }
}
```

```

    if( startA > lenA && startB > lenB ){
        if( min_value > sum )
            min_value = sum;
        return;
    }

    for( int i=0; i < 3; i++ ){
        if(i == 0){
            if( startA <= lenA - 1 ){
                int multi = A[startA] * A[startA+1];
                sum += multi;
                dfs( A, B, startA + 2, lenA, startB, lenB, sum );
                sum -= multi;
            }
        }

        if( i == 1 ){
            if( startA <= lenA && startB <= lenB ){
                int multi = A[startA] * B[startB];
                sum += multi;
                dfs( A, B, startA + 1, lenA, startB + 1, lenB, sum );
                sum -= multi;
            }
        }

        if( i == 2 ){
            if( startB <= lenB - 1 ){
                int multi = B[startB] * B[startB+1];
                sum += multi;
                dfs( A, B, startA, lenA, startB + 2, lenB, sum );
                sum -= multi;
            }
        }
    }
}

int main()
{

    int A[1000], B[1000];
    int combine[2000];
    int lenA = 4, lenB = 4;
    int sum = 0;
    int startA = 0;
    int startB = 0;
    A[0] = 1;

```

```
A[1] =2;
A[2] = 3;
A[3] = 4;

B[0] = 5;
B[1] = 6;
B[2] = 7;
B[3] = 8;

dfs( A, B, startA, lenA -1 , startB, lenB -1, sum );
cout << min_value;

std::cout << "Hello, world!\n";
return 0;
}
```