# DP问题模型：最大子列和问题

## 例题： 1007 Maximum Subsequence Sum (25分)

Given a sequence of $K$ integers { $N_1$, $N_2$, ..., $N_K$ }. A continuous subsequence is defined to be { $N_i$, $N_{i+1}$, ..., $N_j$ } where $1 \leq i \leq j \leq K$. The Maximum Subsequence is the continuous subsequence which has the largest sum of its elements. For example, given sequence { -2, 11, -4, 13, -5, -2 }, its maximum subsequence is { 11, -4, 13 } with the largest sum being 20.

Now you are supposed to find the largest sum, together with the first and the last numbers of the maximum subsequence.

### Input Specification:

Each input file contains one test case. Each case occupies two lines. The first line contains a positive integer $K$ ($\leq$10000). The second line contains $K$ numbers, separated by a space.

### Output Specification:

For each test case, output in one line the largest sum, together with the first and the last numbers of the maximum subsequence. The numbers must be separated by one space, but there must be no extra space at the end of a line. In case that the maximum subsequence is not unique, output the one with the smallest indices $i$ and $j$ (as shown by the sample case). If all the $K$ numbers are negative, then its maximum sum is defined to be 0, and you are supposed to output the first and the last numbers of the whole sequence.

### Sample Input:

```
10
-10 1 2 3 4 -5 -23 3 7 -21
```

### Sample Output:

```
10 1 4
```

### 思路： DP

1. 以前就在陈越姥姥的课上听过这个，知道大概做法。但我依然只能从表面理解，不能想明白他究竟是为什么……

### 参考代码

```
#include <iostream>
#include <vector>
using namespace std;
int main()
{
```

```cpp
    int number = 0;
    scanf("%d", &number);
    vector<int> seq(number);
    int leftIndex = 0, rightIndex = number - 1, sum = -1, temp = 0, tempIndex =
0;
    for(int i=0;i<number;i++)
    {
        scanf("%d", &seq[i]);
        temp += seq[i];
        if(temp<0)
        {
            temp = 0;
            tempIndex = i + 1;
        }else if(temp>sum)
        {
            sum = temp;
            leftIndex = tempIndex;
            rightIndex = i;
        }
    }
    if (sum < 0) sum = 0;
    printf("%d %d %d", sum, seq[leftIndex], seq[rightIndex]);
    return 0;
}
```