

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА СУМИ

Задачата може да се реши като се генерират всички възможни суми и се намери най-голямата от тях. Такова решение (sum-slow.cpp) обаче е със сложност $O(n^2)$ и ще бъде оценено само с 30% от максималния брой точки на задачата.

Решение със сложност $O(n)$:

Нека редицата от числата, записани върху картончетата е $a[0], a[1], \dots, a[n-1]$. Да означим с $\text{maxres}[i]$ най-голямата от разглежданите суми, за които последното събираемо е $a[i]$, а с $\text{maxsum}[i]$ – отговорът на задачата, ако я решаваме само за редицата $a[0], a[1], \dots, a[i]$. Верни са следните твърдения:

```
maxres[0] = maxsum[0] = a[0];
maxres[i] = max{maxres[i-1] + a[i], a[i]} за i = 1, 2, ..., n-1;
maxsum[i] = max{maxsum[i-1], maxres[i]} за i = 1, 2, ..., n-1.
```

Ще отбележим още, че използването на масиви не е необходимо.

```
#include<cstdio>
#include<algorithm>
using namespace std;

int main()
{
    int n, a, maxres, maxsum;
    scanf("%d", &n);
    scanf("%d", &a);
    maxsum = maxres = a;
    for(int i=1; i<n; i++)
    {
        scanf("%d", &a);
        maxres = max(maxres+a, a);
        maxsum = max(maxres, maxsum);
    }
    printf("%d\n", maxsum);
    return 0;
}
```

Автор: Младен Манев