

## 用01背包解决石子归并问题

题目：有一堆石头质量分别为 $W_1, W_2, W_3 \dots W_N$ . ( $W \leq 100000$ )现在需要你将石头合并为两堆，使两堆质量的差为最小。

这道题目可以用01背包问题来解决。即求出和最接近 $\text{sum}/2$ 的一个子集 令 $f(i, j)$ 表示前 $i$ 个元素中和最接近 $j$ 的子集的和（有点绕），则有： $f(i, j) = \max(f(i-1, j), f(i-1, j-a[i])+a[i])$ ，其中 $a$ 数组是用来存储所有石头的质量的。

源码如下：

```
#include <stdio.h>
#define N 5
// Author: 397090770
// E-mail: wyphao.2007@163.com
// 转载请注明。
int do_(int *arr, int m){
    if(arr == NULL){
        return;
    }
    int i = 0, j = 0;
    int c[N + 1][100]; //这个相当于上面的f(i,j)
    for(i = 0; i < N + 1; i++) //初始化
        for(j = 0; j < 100; j++)
            c[i][j] = 0;
    for(i = 1; i < N + 1; i++)
        for(j = 1; j < m + 1; j++){
            if(arr[i - 1] <= j){
                if(arr[i - 1] + c[i-1][j-arr[i - 1]] > c[i-1][j])
                    c[i][j] = arr[i - 1] + c[i-1][j-arr[i - 1]];
            }
            else
                c[i][j] = c[i-1][j];
        }
    else
        c[i][j] = c[i-1][j];
}

//printf("%d\n", c[N][m]);
return c[N][m]; //最后一个就是最优值
}

int main(){
    int arr[N] = {
        /*1, 5, 7, 8, 9, 6, 3, 11, 20, 17*/
    }
}
```

```
5, 8, 13, 27, 14
};
int sum = 0;
int i = 0;
for(i = 0; i < N; i++){
    sum += arr[i];
}
int result = do_(arr, sum / 2);
printf("%d\n", sum - 2 * result);
return 0;
}
```

上面的代码能求出分出来的两堆石头之间质量之差，要输出两堆石头各自包含那几个数据，还需要加许多的代码。本程序的空间复杂度比较高，不过可以优化到只用一维的数组来存。

```
c[i][j] = arr[i - 1] + c[i-1][j-arr[i - 1]];
```

本博客文章除特别声明，全部都是原创！

原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。

本文链接: [【】](#) ( )