

# ZOJ Monthly J ZOJ 3631 Watashi's BG Solution

本作品采用[知识共享署名-非商业性使用-相同方式共享 3.0 Unported 许可协议](#)进行许可

write by Gestalt Lur

2012-07-30

## 题目大意

用  $N(N \leq 30)$  个物品填充体积为  $M(1 \leq M \leq 10^7)$  的背包，问最多能够填充的体积。

## 算法分析

因为  $N$  很少，考虑到各种剪枝，可以用 DFS 来搜索之。首先从大到小给物品排一遍序，然后搜索时可以用如下剪枝：

$M = \min(\text{物品总体积}, M)$ ;

当前解不等于  $M$ ;

当前的体积加上当前搜索位置之后的所有物品体积如劣于最优解，则不必继续；(这也是预处理排序的原因，后面的物品的体积也可以预处理求出)

然后进行满足条件的 DFS 即可。

## 参考代码

```
/*
ZOJ3631
write by Gestalt Lur
2012-07-30
ACCEPTED
*/
#include<algorithm>
#include<cstdio>
#define max(a,b)(a>b?a:b)
#define MAXN 32
using std::sort;

int n , m;
long long ans;
int a[ MAXN ] , rest[ MAXN ];

void dfs( int x , int cnt )
{
    ans = max( ans , cnt );
    ++ x;
    if( cnt + a[ n ] <= m and x <= n and ans not_eq m
        and cnt + rest[ x ] > ans )
    {
        if( a[ x ] + cnt <= m )
            dfs( x , a[ x ] + cnt );
        dfs( x , cnt );
    }
}
```

```

    }
    return ;
}

inline bool cmp( int a , int b )
{ return ( a > b ); }

void init()
{
    ans = 0;
    a[ 0 ] = 0;
    for( int i = 1 ; i <= n ; ++ i )
    {
        scanf( "%d" , &a[ i ] );
        a[ 0 ] += a[ i ];
    }
    sort( a + 1 , a + n + 1 , cmp );
    m = m < a[ 0 ] ? m : a[ 0 ];
    rest[ 0 ] = a[ 0 ];
    for( int i = 1 ; i <= n ; ++ i )
    {
        rest[ i ] = rest[ 0 ];
        rest[ 0 ] -= a[ i ];
    }
    return ;
}

int main()
{
    while( scanf( "%d %d" , &n , &m ) not_eq EOF )
    {
        init();
        dfs( 0 , 0 );
        printf( "%lld\n" , ans );
    }
    return 0;
}

```