**АНАЛИЗ**

Тестовете с *N,M* ≤ 20 могат да бъдат решени с пълно изчерпване със сложност *O(2N + 2M)*.

Тестовете с *N,M* ≤ 300 могат да бъдат решени чрез метода на динамичното програмиране. Намират се всички възможни суми на подмножества на *А* и на подмножества на *B* и се намират 2 от тях с еднаква сума. Това решение има сложност *O(N3 + M3).*

Решението за пълен брой точки, със сложнот *O(N + M)* е следното:  
Започваме да строим двете търсени множества на стъпки. Нека *Ak, Bk* са двете множества след *k* стъпки, като *А0* = {*a0*}, *B0* = {*b0*} и нека означим сумата от елементите на множество *X* с *S(X)*. Строим множествата по следния алгоритъм:  
*dк = S(Aк) – S(Bк)*  
*Аk+1 = Aк*  
*Bk+1 := Bk*  
ако *dк* < 0, към *Ak+1* добавяме първия все още недобавен елемент от *A*.  
ако *dk* > 0 ,към B*k+1* добавяме първия все още недобавен елемент от *B*.  
ако *dk* == 0, връщаме множествата *Ak* и *Bk* и приключваме.

Важно е да се отбележи, че за всяко *k*, *dk* е в отворения интервал (*-n, m*).

Да допуснем, че при изпълнението на алгоритъма *dk* не е ставало 0. Това значи че в някакъв момент е трябвало да добавяме елемент от множество, което вече сме изчерпали. Без ограничение на общността, нека това да бъде множеството *А*. Тъй като *A* има *n* елемента, това значи че можем да намерим *n* на брой стъпки *i1, i2, … in* такива че, за всяко *h* в [1, *n*]:  
1) *Aih* = {*a1, a2, …, ah*} (*А*i­h се състои от първите *h* елемента на *A*)   
2) *dih* = *S*(*Aih*) – *S*(*Bih*) < 0   
 *i1, i2, … in* съществуват, защото сме допуснали че множеството *A* е изчерпаното, което ни гарантира че *in* съществува и *S(Ain) – S(Bin) <* 0, но от съществуването на *in* следва съществуването на *in-1* и по индукция следва и съществуването на всички останали *i1, i2, … in-2 .*  
  
 Понеже за всяко *h* от 1 до *n, dih* е в отворения интервал (-*n*, 0) от принципа на Дирихле можем да заключим, че съществуват *p* и *q*, с *dip* = *diq*  и *p* < *q*. Очевидно е, че *Aip* е подмножество на *Aiq* и също така   
*Bip* e подмножество на *Biq*. Следователно:  
*S(Aiq) = S(Aip) + S(Aiq \ Aip)* и *S(Biq ) = S(Bip) + S(Biq \ Bip) =>  
S(Aiq) - S(Biq) = S(Aip) + S(Aiq \ Aip) - S(Bip) - S(Biq \ Bip)* =>  
*S(Aiq) - S(Biq) = S(Aip) - S(Bip) + S(Aiq \ Aip) - S(Biq \ Bip)* =>  
*diq = dip + S(Aiq \ Aip) - S(Biq \ Bip)* =>  
*S(Aiq \ Aip) - S(Biq \ B\_ip) = diq - dip =* 0  
Можем да заключим че множествата *(Aiq \ Aip*)и *(Biq \ Bip)* имат еднаква сума.

*Автор: Антъни Господинов*