分析：

如果数组总和sum为odd, 一定不可行。

如果是总和sum是even，或许可行。

当总和是even时候，问题就变成了，在数组中的元素的子集中，能否构成sum/2.

对此有两种方法，

方法一： 用递归，

public class Solution {

public boolean canPartition(int[] nums) {

//Method 1 recursion

int sum=0;

for(int i=0;i<nums.length;i++)

{

sum=sum+nums[i];

}

if(sum%2!=0)

return false;

else

return recursion(sum/2,nums,0,nums.length-1);

}

public boolean recursion(int sum,int[] nums,int start,int end)

{

boolean results=false;

if(sum==0)

return true;

if(sum<0)

return false;

else

{

for(int i=start;i<=end;i++)

{

swap(nums,start,i);

results=results||recursion(sum-nums[start],nums,start+1,end);

swap(nums,start,i);

}

return results;

}

}

public void swap(int[] nums, int i, int j)

{

int temp=nums[i];

nums[i]=nums[j];

nums[j]=temp;

}

}

方法二：dynamic programing

class Partition {

// Returns true if arr[] can be partitioned in two subsets of

// equal sum, otherwise false

static boolean findPartition (int arr[], int n)

{

int sum = 0;

int i, j;

// Caculcate sun of all elements

for (i = 0; i < n; i++)

sum += arr[i];

if (sum%2 != 0)

return false;

boolean part[][]=new boolean[sum/2+1][n+1];

// initialize top row as true

for (i = 0; i <= n; i++)

part[0][i] = true;

// initialize leftmost column, except part[0][0], as 0

for (i = 1; i <= sum/2; i++)

part[i][0] = false;

// Fill the partition table in botton up manner

for (i = 1; i <= sum/2; i++)

{

for (j = 1; j <= n; j++)

{

part[i][j] = part[i][j-1];

if (i >= arr[j-1])

part[i][j] = part[i][j] ||

part[i - arr[j-1]][j-1];

}

}

/\* // uncomment this part to print table

for (i = 0; i <= sum/2; i++)

{

for (j = 0; j <= n; j++)

printf ("%4d", part[i][j]);

printf("\n");

} \*/

return part[sum/2][n];

}

/\*Driver function to check for above function\*/

public static void main (String[] args)

{

int arr[] = {3, 1, 1, 2, 2,1};

int n = arr.length;

if (findPartition(arr, n) == true)

System.out.println("Can be divided into two "

"subsets of equal sum");

else

System.out.println("Can not be divided into"

" two subsets of equal sum");

}

}

/\* This code is contributed by Devesh Agrawal \*/

Output:

Can be divided into two subsets of equal sum

Following diagram shows the values in partition table. The diagram is taken form the wiki page of partition problem.

