**★★★★☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：11234: Halloween treats**

**解題者：孫世諭**

**解題日期：2024年5月9日**

**題意：**

給定一個 c 值，代表小朋友的數量；再給定 n 個值 (n1, n2, n3 …… ni)，代表每個鄰居所給予的糖果數量，其中，一個鄰居只能給一次糖果並全部送出，且
(1 ≤ c ≤ n ≤ 100000)。小朋友必須找出一個鄰居糖果的子集合，總和能夠被 c 整除。如果答案有多個，只需隨意輸出其中一個；如果沒有答案，輸出no sweets。

**題意範例：**

4 5
1 2 3 7 5 -> 3 5

3 6
7 11 2 5 13 17 -> 2 3 4

**解法：**

因為條件為 1 ≤ c ≤ n ≤ 100000，所以使用 Prefix Sum + 鴿籠原理。先將所有的給定的值做 Prefix Sum 並取c的餘數，如 𝑝\_1 𝑚𝑜𝑑 𝑐, 𝑝\_2 𝑚𝑜𝑑 𝑐 ……𝑝\_𝑖 𝑚𝑜𝑑 𝑐，接下來觀察擁有相同的值或是為0的餘數，兩者相同值的中間區間即是答案。

**解法範例：**

以第一個例子為例：1 2 3 7 5:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Index | 1 | 2 | 3 | 4 | 5 |
| 原本的輸入 | 1 | 2 | 3 | 7 | 5 |
| Prefix Sum | 1 | 3 | 6 | 13 | 18 |
| 取 4 的餘數 | 1  | 3 | 2 | 1 | 2 |

取餘數相同代表意義如下：

 13 = 1 + 2 + 3 + 7 取餘數為 1
 -) 1 = 1 取餘數為 1
 ---------------------------------------------------------
 2 + 3 + 7 取餘數為 0 可以整除
因此，其中一個答案會是 2 3 4。

**討論：**

1. 可以發現，使用 Prefix Sum 會有問題：
這個解答必定會是一個連續的區間，會直接忽略不連續的區間，如 1 2 3 7 5。
但是根據條件 1 ≤ c ≤ n ≤ 100000，並且0 ≤ 𝑛\_𝑖 𝑚𝑜𝑑 𝑐 ≤ 𝑐−1，而又根據鴿籠原理可以知道，在c ≤ n 的情況下，必定存在兩個以上𝑛\_𝑖 𝑚𝑜𝑑 𝑐 相同。所以我們完全不需要在乎不連續區間的情況，因為一定會有解。時間複雜度為 O(n)。

2. 如果題目不是限制 1 ≤ c ≤ n ≤ 100000，則可以使用 Dynamic Programming 解決 Sum of Subset 的問題方式來處理本題目，時間複雜度為 O(𝑛^2)，在這題會TLE。

**程式：**

#include <iostream>

#include <cstring>

#include <algorithm>

#include <unordered\_map>

#include <vector>

using namespace std;

unordered\_map<int, int> p;

int main()

{

 int c, n;

 while(true)

 {

 cin >> c >> n;

 if(c+n==0) break;

 vector<int> a(n+1,0);

 vector<int> vis(n+1,-1);

 p.clear();

 for (int i=1; i<=n; ++i) cin >> a[i];

 vis[0] = 0;

 p[0] = 0;

 for (int i = 1; i <= n; i++)

 {

 p[i]=(p[i-1]+a[i])%c;

 if (vis[p[i]] == -1) vis[p[i]] = i;

 else

 {

 for (int j=vis[p[i]]+1; j<i; j++) cout << j << ' ';

 cout << i << endl;

 break;

 }

 }

 }

 return 0;

}