**★★★☆☆**

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

**題號：10954: Add All**

**解題者：林玟妗**

**解題日期：2024年3月21日**

**題意：**

將做加法要付出的代價（cost） 定義為這2個數的總和，給予 N 個數字，找出把 N 個數加起來使得付出的代價最少。

**題意範例：**

1 2 3 → 9
 1 2 3 4 → 19

**解法：**

此做法時間複雜度為 O(n^2 logn)，並非好的演算法，可以用heap或pre-order queue的方式來解。

**解法範例：**

66 32 44 70 5

 🡺 5 32 44 66 70 ;cost=0//sort

 🡺 37 44 66 70 ;cost=37 //v[0]+v[1],cost+=v[0],刪去v[1]

 🡺 81 66 70 ;cost=118 //v[0]+v[1],cost+=v[0],刪去v[1]

 🡺 66 70 81 ;cost=118 //sort

 🡺 136 81 ;cost=254 // v[0]+v[1],cost+=v[0],刪去v[1]

 🡺 81 136 ;cost=254 //sort

 🡺 217 ;cost=471 //v[0]+v[1],cost+=v[0],刪去v[1]

 🡺 cost=471

**討論：**

因cost仍須加進字串裡，當所得cost大於v[1]時，會造成字串非升冪排列，使後續數字相加時不為最小兩數相加，導致所得cost不為最小，故需再進行一次sort。

**程式：**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main(){

 int num;

 while(cin>>num){

 if(num==0)

 break;

 vector<int> val; //數字字串

 long long int cost=0; //cost之和

 int n;

 for(int i=0;i<num;i++){

 cin>>n;

 val.push\_back(n); //儲存數字字串

 }

 sort(val.begin(),val.end()); //排序

 while(val.size()>1){ //將數字相加直至最後一個數

 val[0]+=val[1];

 val.erase(val.begin()+1); //去除已相加的數

 cost+=val[0]; //累加cost

 if(val[0]>val[1]) //重新排序字串

 sort(val.begin(),val.end());

 }

 cout<<cost<<endl;

 }

 return 0;

}