

# 2023 ITSA 程式線上自我評量 5 月場題目解析

國立成功大學資訊工程學系 陳俊安

## Problem 2. 公平

(Time Limit: 1 seconds)

### 問題描述:

縣長剿完匪後，剿獲了一堆物資，準備將物資歸還給縣民，但因物品主已經不可考，縣長決定先將這些物品分成兩堆後再分派給東西兩區，再往下分配，每個物品均有其標價且不可分割。縣長要求的就三件事：『公平、公平、還是公平』，現在請您將這些物品分成兩堆，使這兩堆總價值差距最小。

### 輸入說明

輸入第 1 行為 1 個正整數  $k$ ， $1 \leq k \leq 20$ ，代表共有  $k$  件物品，接著有其價值分別為  $n_1, n_2, \dots, n_k$ ， $1 \leq n_i \leq 1000$ ，而此  $k$  個正整數間以空格隔開。

### 輸出說明:

輸出兩堆價值差距的最小值。

### 範例

Sample1 Input:	Sample1 Output:
5 5 13 21 30 35	2

# 本題會介紹三種作法

- 枚舉 (遞迴版)
- 枚舉 (位元版)
- 動態規劃

# 遞迴作法

```
1  #include <iostream>
2  #include <cmath>
3
4  #define int long long
5
6  using namespace std;
7
8  int a[20],ans = 1e9,n;
9
10 void dfs(int idx,int sum1,int sum2)
11 {
12     if( idx == n )
13     {
14         ans = min(ans,abs(sum1-sum2));
15         return;
16     }
17
18     dfs(idx+1,sum1+a[idx],sum2);
19     dfs(idx+1,sum1,sum2+a[idx]);
20 }
21 signed main()
22 {
23     cin >> n;
24     for(int i=0;i<n;i++) cin >> a[i];
25     dfs(0,0,0);
26
27     cout << ans << "\n";
28 }
```

# 位元作法

```
1  #include <iostream>
2  #include <vector>
3
4  #define int long long
5
6  using namespace std;
7
8  signed main(void)
9  {
10     int n,total = 0;
11     cin >> n;
12     vector<int> a(n);
13     for(int i=0;i<n;i++) cin >> a[i] , total += a[i];
14
15     int ans = 1e9;
16
17     for(int i=0;i<(1<<n);i++)
18     {
19         int sum = 0;
20         for(int k=0;k<n;k++)
21         {
22             if( ( i & ( 1 << k ) ) != 0 ) sum += a[k];
23         }
24
25         int sum2 = total - sum;
26         ans = min(ans,abs(sum-sum2));
27     }
28
29     cout << ans << "\n";
30
31     return 0;
32 }
```

# 動態規劃作法

```
1  #include <iostream>
2
3  using namespace std;
4
5  int a[21],dp[21][20001];
6
7  int main(void)
8  {
9      int n,sum = 0;
10     cin >> n;
11     for(int i=1;i<=n;i++) cin >> a[i] , sum += a[i];
12     for(int i=1;i<=n;i++)
13     {
14         for(int k=0;k<=sum/2;k++)
15         {
16             if( k - a[i] >= 0 ) dp[i][k] = max( dp[i-1][k] , dp[i-1][k-a[i]] + a[i] );
17             else dp[i][k] = dp[i-1][k];
18         }
19     }
20
21     cout << ( sum - dp[n][sum/2] ) - dp[n][sum/2] << "\n";
22 }
```

額外補充：折半枚舉

## Problem 3. 商品比重

(Time Limit: 1 seconds)

### 問題描述:

大北生活廣場剛進貨了大批的商品，這些商品有分清潔用品、零食、五金等眾多類型不等，今該店長在整理時，想知道各種商品所占全部商品之比重，以便於下次進貨的依據，避免商品太偏重於某個類型。

### 輸入說明

輸入第 1 行為 1 個正整數  $k$  ( $1 \leq k \leq 20$ ) 表示有  $k$  種商品，接著有  $k$  行，每行先輸入以 A~Z 大寫英文字母表示商品名稱，接著輸入該類型商品總數  $m$  ( $1 \leq m \leq 1,000$ )，文字與數字中間以一個空白隔開。

### 輸出說明:

輸出每個商品所占全部商品之比重，比重最多表示到小數點 2 位(四捨五入)。

按照英文字母順序 A~Z 依序輸出。

※有部份程式語言提供之四捨五入 function 會出錯，建議自行寫一支四捨五入 function。



# 實作

```
10 void init()
11 {
12     for(int i=0;i<26;i++) w[i] = -1;
13     return;
14 }
15 double solve(double num)
16 {
17     num *= 1000;
18     if( (int)num % 10 >= 5 ) num /= 10 , num += 1 , num /= 100;
19     else num /= 1000;
20
21     return num;
22 }
23 signed main(void)
24 {
25     int n,sum = 0;
26     cin >> n;
27
28     init();
29     for(int i=0;i<n;i++) cin >> id[i] >> w[ id[i] - 'A' ] , sum += w[ id[i] - 'A' ];
30     for(int i=0;i<26;i++)
31     {
32         if( w[i] != -1 )
33         {
34             double ans = (double)w[i] / (double)sum;
35
36             ans = solve(ans);
37             ans *= 100;
38             ans = (int)ans;
39             ans /= 100;
40
41             cout << fixed << setprecision(2) << (char)('A'+i) << "(" << ans << ")"\n";
42         }
43     }
44     return 0;
45 }
```

## Problem 4. 彈珠

成績: 0 / 倒扣: 0.8

## Problem 4. 彈珠

(Time Limit: 1 seconds)

### 問題描述:

小強在 2 樓樓梯旁玩耍彈珠，不小心把 1 顆彈珠滾到 1 樓了，彈珠掉下去時，有可能 1 次掉 1 階樓梯，也有可能 1 次掉 2 階樓梯，請問彈珠掉到 1 樓的過程中，不同的掉落方法有幾種？

### 輸入說明

第 1 行輸入 1 正整數  $N(1 \leq N \leq 10)$  表示共有  $N$  筆測試資料。

每筆測試資料為 1 正整數  $M(1 \leq M \leq 20)$  表示 2 樓到 1 樓共有  $M$  階樓梯。

### 輸出說明:

輸出彈珠掉下去共有幾種不同的方法之總數。

# 實作

```
1  #include <iostream>
2  #define int long long
3
4  using namespace std;
5
6  int f[21];
7
8  signed main()
9  {
10     f[0] = 1 , f[1] = 1;
11     for(int i=2;i<=20;i++) f[i] = f[i-1] + f[i-2];
12
13     int q;
14     cin >> q;
15     while(q-->0)
16     {
17         int n;
18         cin >> n;
19         cout << f[n] << "\n";
20     }
21 }
```

## 額外補充：矩陣快速冪 & Top-Down