Performance Measurement (A+B)

张远帆

Date:2024-03-09

Chapter 1:Introduction

Problem Description:This problem requires us to find two integers from N given numbers as their sum equals to c.We have to find all the numbers suitable for the problem.Two different algorithms are required,and we have to measure the time for the program to run.

Algorithm Background description:For algorithm 1,we just use enumeration method.For algorithm 2,we use Bucket sorting to find the number.

Chapter 2: Algorithm Specification

Algorithm 1: Using a double loop, enumerate each a [i] and a [j]. If their sum is c, output two numbers. If not found, output nofound.

pseudo code:

begin

input n,c,a[i]

for I 0:10

for j 0:10

if a[i]+=a[j]=c then output

end

Data structure:array.

Algorithm 2: Using bucket sorting, that is, creating an array where the value a [i] indicates whether i has appeared, 1 indicates it has appeared, and 0 indicates it has not appeared.

pseudo code:

begin

input n,c,a[i]

b[a[i]]=1

if b[c-a[i]]=1 then output a[i],c-a[i]

end

Data structure:array

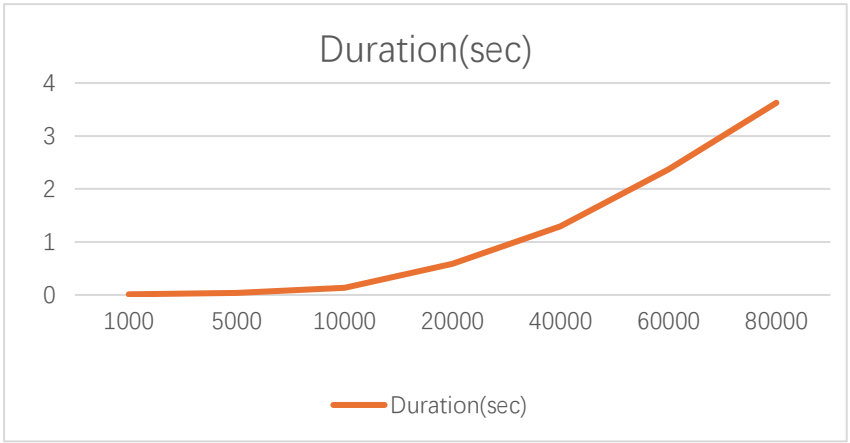
Chapter 3:Testing Result

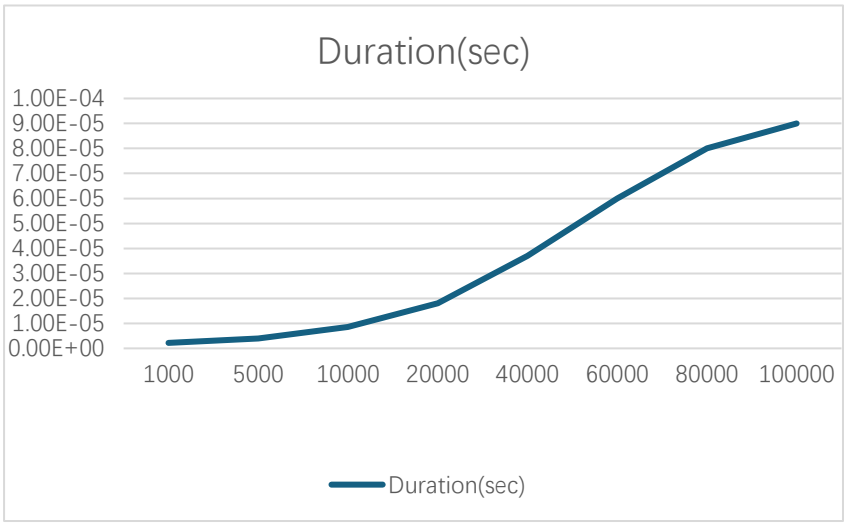
Test case 1:Normal numbers ranging from 1 to n,while v=10000

Expected result:output every numbers suitable.

Actual behavior:run as expected.

V=10000	N	1000	5000	10000	20000	40000	60000	80000	100000
Algorithm 1	Iterations(K)	50	2	1	1	1	1	1	1
	Ticks	24	23	41	137	587	1296	2371	3626
	Total Time(sec)	0.024	0.023	0.041	0.137	0.587	1.296	2.371	3.626
	Duration(sec)	0.0004	0.012	0.041	0.137	0.587	1.296	2.371	3.626
Algorithm 2	Iterations(K)	10000	5000	5000	1000	1000	500	500	500
	Ticks	22	20	43	18	37	29	41	49
	Total Time(sec)	0.022	0.02	0.043	0.018	0.037	0.029	0.041	0.049
	Duration(sec)	2.20E-06	4.00E-06	8.60E-06	1.80E-05	3.70E-06	6.00E-05	8.00E-05	9.00E-05





Status:Pass

Test case 2:There is no answer to the question,that is,there should be no output.The purpose of the test case is to test if the program could out put nofound.

The program run as expected result.

Status:Pass

Test case 3:The worst condition,that v=100000,the numbers are given randomly,and the same number could appear one more time,to test the maximum time the algorithm need to run.

The program run as expected result.

Status:Pass

Chapter 4 Analysis and Comments

Algorithm 1: An i * j loop was used, and the upper limits of i and j are both n, so the time complexity of this algorithm is O (n^2). An array of length n was used to store data, but only one space was exchanged during the subsequent comparison process, resulting in a

spatial complexity of O (n)

Time complexity:O(n^2) Space complexity: O(n)

Algorithm 2: Conducted n loops to input and search for data, resulting in a time complexity of O (n).An array of length n was used to store how many times each number appeared, resulting in a spatial complexity of O (n)

Tine complexity:O(n) Space complexity:O(V)

Appendix:Sourcre code in C

Algorithm 1

```
 #include<stdio.h>
int main()
{
    int n,a[100001],c;//define n,c and a
    scanf("%d %d",&n,&c);//read n and c
    for(int i=0;i<n;i++)
    scanf("%d",&a[i]);//输入 a
    int flag=0;//flag is used to mark if the numbers are found
    for(int i=0;i<n;i++)//Traverse the combination of all array elements
    for(int j=i+1;j<n;j++)
    {
        if(a[i]+a[j]==c)//if the sum of the numbers equals to c
```

```
            {
                printf("%d %d\n",a[i],a[j]);//print a[i] and a[j]
                flag=1;//mark flag as 1,meaning the numbers are found
                break;
            }
            if(flag==1)break;
        }
        if(flag==0)printf("No Found\n");//if didn't find the number,print nofound
        return 0;
}
```

Algorithm 2

```
#include<stdio.h>
int n,c;
int a[100001]={0},b[100001];//array a is used to mark if the number has appeared
//for example,if a[i]=1,means number i is in the array b
//array b is used to store the n given numbers
//suppose the integers are no larger than 100000
int main(void)
{
        scanf("%d %d",&n,&c);//read n and c
        for(int i=0;i<n;i++)
        {
            scanf("%d",&b[i]);//read the numbers
```

```
            if(a[c-b[i]]==1)printf("%d %d\n",b[i],c-b[i]);//if c-b[i] is appeared,than print the
two numbers

            a[b[i]]=1;//mark array a[b[i]] as 1

        }

        return 0;

}
```

Declaration:I hereby declare that all the work done in this project titled " **Performance Measurement (A+B)**" is of my independent effort.