Комбинаторика Порождение чисел Фибоначчи/Разбиение числа

Code by Xumingchuan

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <time.h>

#define N 5

int Frr[N];
int Par[N];

int min(int A,int B){
    if(A<=B) return A;
    return B;
}

//Finbonaqii
void FrrInit(){
    Frr[0]=1;
    Frr[1]=1;
    for(int i=2;i<N;i++) Frr[i]=-1;
}

int Fibo(int M){
    if(M<=1) return 1;
    if(Frr[M]<0) Frr[M]=Fibo(M-1)+Fibo(M-2);
    return Frr[M];
}

//Partition (with repeat)
int Partition(int NN){
    if(NN<=1) return 1;
    int S=0;
    for(int i=1;i<=NN;i++)
        S+=Partition(NN-i);
    return S;
}

//PartitionPro  (reduce recursion)
void ParInit(){
    Par[1]=1;
    for(int i=2;i<=N;i++) Par[i]=-1;
}

int PartitionPro(int NN){
    if(NN<=1) return 1;
    if(Par[NN]<0){
        Par[NN]=0;
        for(int i=1;i<=NN;i++)
        Par[NN]+=PartitionPro(NN-i);
    }
    return Par[NN];
}

//Partition (without repeat)
//M - number ,Max - biggest adder in M
int Parti(int M,int Max){
    if(M<=1) return 1;
    int S=0;
    for(int i=1;i<=min(M,Max);i++)
        S+=Parti(M-i,i);
    return S;
}

int main(){
    printf("Number:32\n");
    printf("\n");

    //Finbonaqii
    printf("Finbonaqii number:\n");
    FrrInit();
    for(int i=2;i<N;i++)
      Frr[i]=Fibo(i);
    for(int j=0;j<N;j++)
      printf("%d ",Frr[j]);
    printf("\n");
    printf("\n");
```

```
77
78      //Partition of number N(repeat)(recursion)
79      double Start1=clock();
80      printf("Partition():\n");
81      printf("%d\n",Partition(N));
82      double End1=clock();
83      printf("RunTime:%f\n",(End1-Start1)/CLOCKS_PER_SEC);
84      printf("\n");
85
86      //Partition of number N(reduce recursion)
87      double Start2=clock();
88      ParInit();
89      printf("PartitionPro():\n");
90      for(int i=0;i<=N;i++)
91        Par[i]=PartitionPro(i);
92      printf("%d",Par[N]);
93      printf("\n");
94      double End2=clock();
95      printf("RunTime:%f\n",(End2-Start2)/CLOCKS_PER_SEC);
96      printf("\n");
97
98      //Parti of number N
99      printf("Parti():\n");
100     printf("%d",Parti(N,N));
101     return 0;
102 }
```

ВЫХОД:

```
Number:30

Finbonaqii number:
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 51
4229 832040

Partition():
536870912
RunTime:3.205000

PartitionPro():
536870912
RunTime:0.001000

Parti():
5604
--------------------------------
```

FrrInit( ) – Инициализируем массив, который используется для хранения числа Фибоначчи, соответствующего каждому входному номеру M. Когда функция вызывается в рекурсии, она напрямую обращается к соответствующему значению в массиве, чтобы уменьшить количество вызовов.

Fibo( ) – Вычислить M-е число Фибоначчи

Partition( ) – Простое разбиения числа(с повторения)

ParInit( ) – Аналогично FrrInit( ) для разбиения числа

PartitionPro( ) – Разбиение числа, используя массив(с повторения)

Parti( ) – Разбиение числа без повторения, в которую входится дополнительный параметр – максимальное возможное слагаемое

В функции main вычисляется время выполнения программы без использования и использования массивов, и можно обнаружить, что использование массива для хранения данных с целью уменьшения рекурсии значительно сокращает время выполнения программы.

В результате заменился входное число на 30