数据结构实验八 递归的实现

[ 发布人: <u>易法令</u> 截止时间: 2021-11-11 23:59:59 ]

- 一、实验目的
- 1、掌握递归原理
- 2、掌握一些常用问题的递归算法设计
- 二、实验内容
- 1.编写一个连加运算递归函数 ADD(n),实现从 1+2+……+n。
- 2. 编写折半查找算法的递归实现和非递归实现。

提示:将要查找的元素 key 与查找区间正中元素相比,若 key 小,则查找区间缩小至前半部份查找,若 key 大,则查找区间缩小至后半部份查找;再取其中值比较,每次缩小 1/2 的范围,直到查找成功或失败为止。如递归实现,考虑函数的参数应有哪些。在用循环结构实现时,函数的参数有什么变化?

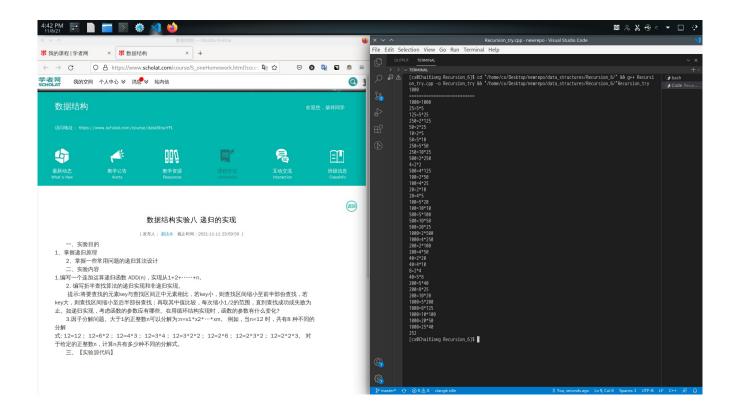
3.因子分解问题。大于 1 的正整数 n 可以分解为:n=x1\*x2\*…\*xm。 例如,当 n=12 时,共有 8 种不同的分解式: 12=12; 12=6\*2; 12=4\*3; 12=3\*4; 12=3\*2\*2; 12=2\*6; 12=2\*3 \*2; 12=2\*3。 对于给定的正整数 n,计算 n 共有多少种不同的分解式。

## 三、【实验源代码】

```
#include <bits/stdc++.h>
using namespace std;
const int N = 100000;
int a[N];
int aaa = 0;
int resolve(int n) {
   int sum = 1;
   aaa++;
   // cout << "----" << aaa << "--n:" << n << endl;
   if (n < N && a[n] != 0) {
     return a[n];
   }
   for (int i = 2; i <= sqrt(n); i++) {
     if (n % i == 0) {</pre>
```

```
if (i * i == n) {
     sum += resolve(i);
     cout << n << "=" << i << "*" << i << endl;
   } else {
     sum += resolve(i) + resolve(n / i);
     cout << n << "=" << i << "*" << n / i << endl;
   }
  }
 if (n < N) {
  a[n] = sum;
 return sum;
}
int main() {
 int n;
 cin >> n;
 memset(a, 0, sizeof(a));
 cout << "========" << endl;
 cout << n << "=" << n << endl;
 cout << resolve(n) << endl;</pre>
 return 0;
```

## 四、【实验结果】



## 五、【实验心得】



SCHOLAT.com 学者网

免责声明 | 关于学者网