数据结构

2022

实验报告

实验项目名称：期中

班级：2021级6班

学号：2021302181138

姓名：伍旺旺

指导教师：沈志东

实验时间：2022年4月

数据结构实验期中

一、实验要求

实验一、递归求在给定二叉树结点总数 N 的情况下，二叉树可能拥有的形状数 M

实验二、非递归求在给定二叉树结点总数 N 的情况下，二叉树可能拥有的形状数 M

二、实验环境

硬件：微型计算机

软件：Windows 操作系统、Microsoft Visual Studio Code

三、实验步骤及思路

实验一、递归

- 题目分析
  实验所需递归函数原型为

```
void arrange(int arr[],int idx,int N,int &tree_count)
```

- 实验具体步骤
  1. 递归函数编写：

```
void arrange(int arr[],int idx,int n,long long &tree_count,long long &height)
{
        if(idx == n){
                int w = 1;

                tree_count++;
                printf("   %lld:",tree_count);
                printf("%d",arr[0]);
                for(int i = 1;i <= n - 1;i++){
                        if(arr[i] >= arr[i-1]*2)
                                w++;
                        printf(",%d",arr[i]);
                }
                printf("\n");
                height += w;
                return ;
        }
    int t = arr[idx-1];
        for(int i = idx-1;i >= 0;i--){
                if(arr[i]*2+1 < t)
                        break;
                if(arr[i]*2>t){
                        arr[idx] = arr[i]*2;
                        arrange(arr,idx+1,n,tree_count,height);
                }
                if(arr[i]*2 + 1 > t){
                        arr[idx] = arr[i]*2 + 1;
                        arrange(arr,idx + 1,n,tree_count,height);
                }
        }

}
```

2. main函数:

```
int main()
{
        int n;

        printf("请输入结点个数: ");
        scanf("%d",&n);
        long long count = 0,H = 0;
        a[0] = 1;
        arrange(a,1,n,count,H);
        printf("\ntree_count is %lld when N is %d\n\n",count,n);
        long long t = 1;
        for(int i = n+1;i <= n*2;i++)
                t = 1ll*i*t;
        for(int i = 1;i<=n+1;i++)
                t = t / (1ll*i);
        printf("count = %lld , Catalan = %lld \n\n",count,t);
        printf("average_Height = %.6lf\n\n",H*1.0/count);
        printf("(log2(N)+N)/2 = %.6lf\n\n",(log2(n)+n)*1.0/2);

        return 0;
}
```

实验二、非递归

- 题目分析
  非递归函数原型:

```
void buildtree(int N,int &tree_count)
```

- 实验具体步骤
  1. 非递归函数编写

```
void buildtree(int N,long long &tree_count)
{
        int arr[32],pos[32];
        int idx = 1;

        arr[idx++] = 1;
        pos[1] = 1;
        ST[++top] = 2;
        ST[++top] = 3;
        while(idx > 0){
                int now = ST[top];
                if(idx == N){
                        top--;
                        while(arr[idx-1] < now){
                                tree_count++;
                                printf("  %lld:",tree_count);
                                for(int i = 1;i < idx;i++)
                                        printf("%d,",arr[i]);
                                printf("%d\n",now);
                                now = ST[top];
                                top--;
                        }
                        top++;
                        idx--;
                        continue;
                }
                if(arr[idx - 1] < now){
                        top--;
                        arr[idx] = now;
                        pos[idx] = pos[idx-1];
                        for(int i = pos[idx];i <= idx;i++){
                                if(arr[i]*2+1<arr[idx])
                                {
                                        pos[idx] = i;
                                        continue;
                                }
                                if(arr[i]*2 > arr[idx])
                                        ST[++top] = arr[i]*2;
                                if(arr[i]*2+1 > arr[idx])
                                        ST[++top] = arr[i]*2 + 1;
                        }
                        idx++;
                }
                else idx--;
        }
}
```

2. main函数编写：

```
int main() {
        int n;

        long long e = 0;
        printf("请输入结点个数: ");
        scanf("%d",&n);
        buildtree(n,e);
        printf("\ntree_count is %lld when N is %d\n",e,n);

        return 0;
}
```

四、实验结果及分析

- **实验一、递归**
  运行结果如下:

选择 C:\Windows\system32\cmd.exe

```
请输入结点个数：5
   1:1,2,4,8,16
   2:1,2,4,8,17
   3:1,2,4,8,9
   4:1,2,4,9,18
   5:1,2,4,9,19
   6:1,2,4,5,10
   7:1,2,4,5,11
   8:1,2,4,5,8
   9:1,2,4,5,9
   10:1,2,5,10,20
   11:1,2,5,10,21
   12:1,2,5,10,11
   13:1,2,5,11,22
   14:1,2,5,11,23
   15:1,2,3,6,12
   16:1,2,3,6,13
   17:1,2,3,6,7
   18:1,2,3,7,14
   19:1,2,3,7,15
   20:1,2,3,4,8
   21:1,2,3,4,9
   22:1,2,3,4,6
   23:1,2,3,4,7
   24:1,2,3,4,5
   25:1,2,3,5,10
   26:1,2,3,5,11
   27:1,2,3,5,6
   28:1,2,3,5,7
   29:1,3,6,12,24
   30:1,3,6,12,25
   31:1,3,6,12,13
   32:1,3,6,13,26
   33:1,3,6,13,27
   34:1,3,6,7,14
   35:1,3,6,7,15
   36:1,3,6,7,12
   37:1,3,6,7,13
   38:1,3,7,14,28
   39:1,3,7,14,29
   40:1,3,7,14,15
   41:1,3,7,15,30
   42:1,3,7,15,31

tree_count is 42 when N is 5

count = 42 , Catalan = 42

average_Height = 3.928571
```

- **实验二、非递归**

  运行结果如下:

选择 C:\Windows\system32\cmd.exe

```
请输入结点个数：5
  1:1, 3, 7, 15, 31
  2:1, 3, 7, 15, 30
  3:1, 3, 7, 14, 29
  4:1, 3, 7, 14, 28
  5:1, 3, 7, 14, 15
  6:1, 3, 6, 13, 27
  7:1, 3, 6, 13, 26
  8:1, 3, 6, 12, 25
  9:1, 3, 6, 12, 24
  10:1, 3, 6, 12, 13
  11:1, 3, 6, 7, 15
  12:1, 3, 6, 7, 14
  13:1, 3, 6, 7, 13
  14:1, 3, 6, 7, 12
  15:1, 2, 5, 11, 23
  16:1, 2, 5, 11, 22
  17:1, 2, 5, 10, 21
  18:1, 2, 5, 10, 20
  19:1, 2, 5, 10, 11
  20:1, 2, 4, 9, 19
  21:1, 2, 4, 9, 18
  22:1, 2, 4, 8, 17
  23:1, 2, 4, 8, 16
  24:1, 2, 4, 8, 9
  25:1, 2, 4, 5, 11
  26:1, 2, 4, 5, 10
  27:1, 2, 4, 5, 9
  28:1, 2, 4, 5, 8
  29:1, 2, 3, 7, 15
  30:1, 2, 3, 7, 14
  31:1, 2, 3, 6, 13
  32:1, 2, 3, 6, 12
  33:1, 2, 3, 6, 7
  34:1, 2, 3, 5, 11
  35:1, 2, 3, 5, 10
  36:1, 2, 3, 5, 7
  37:1, 2, 3, 5, 6
  38:1, 2, 3, 4, 9
  39:1, 2, 3, 4, 8
  40:1, 2, 3, 4, 7
  41:1, 2, 3, 4, 6
  42:1, 2, 3, 4, 5
  43:1, 2, 3, 4, 1857421320
  44:1, 2, 3, 4, 2040
  45:1, 2, 3, 4, 4223176

tree_count is 45 when N is 5
```