《算法分析与设计》

实 验 报 告

学 号

姓 名

年 级 2020

专 业 软件工程

院 系 计算机与人工智能学院

二0二二年三月

# 实验1.3程序执行过程分析

1. 实验目的

|  |
| --- |
| 1. 正确理解算法的执行过程 |

1. 实验任务

|  |
| --- |
| 1. 分析下面的算法在样例输入情况下的执行过程及结果 |

1. 实验环境

|  |
| --- |
| * 1. 硬件环境  1. 计算机：拯救者R7000P 2020H 2. CPU: AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz 3. RAM：16GB    1. 软件环境 4. 操作系统：Windows11家庭中文版 5. 开发工具：Visual Studio Code |

1. 实验步骤及结果

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| * 1. 实验预习      1. 分析下面的算法在样例输入情况下程序的执行过程及结果  1. 该算法为寻找一个二维数组中不同行不同列的相同符号的组数。   在输入样例中，n=3,因此创建了一个3\*3的二维数组，其中符号为：  #.#  ###  .#.  又，k=2，即寻找该数组中在不同行不同列的2个为一组的“#”有多少组。算法主体为dfs函数，首先由第一行开始，依次首先执行dfs（1，2），在函数体内，需判断标志数组sign[]是否为0，以及该行该列字符是否为“#”，易知第一行第一列符合，接着递归调用，即执行dfs（2，1），即开始判断第二行是否有符合的字符，此时只需再另寻一个“#”即可，因此k=k-1，此时判断sign[j]时，因为上文已经把sign[1]置1，也即第一列存在“#”号，所以进入sign[2]判断，即判断第二列是否有符合的字符，由样例可知，第二列存在符合条件的字符，所以k=k-1，再次递归时进入k==0判断，数量ans+1。  （2）后续也即按照上述流程进行，最后得出ans=8，也即该二维数组中共有8组个数为2，字符为“#”的字符对。   * + 1. 程序流程描述   如图1所示。  wps  图1 程序流程图   * + 1. 程序代码   #include<cstdio>  #include<cstring>  #include<iostream>  using namespace std;  char chess[10][10];  int sign[10];  int n, k, ans;  void dfs(int *x*, int *k*)  {      if (*k* == 0)      {          ans++;          return;      }      if (*x* > n)          return;      for (int i = *x*; i <= n; i++)      {          for (int j = 1; j <= n; j++)          {               if (!sign[j] && chess[i][j] == '#')              {                  sign[j] = 1;                  dfs(i + 1, *k* - 1);                  sign[j] = 0;              }          }      }  }  int main()  {      while (cin>> n>> k)      {          memset(chess, 0, sizeof(chess));          memset(sign, 0, sizeof(sign));          if (n == -1 || k == -1)              break;          for (int i = 1; i <= n; i++)              for (int j = 1; j <= n; j++)                  cin>> chess[i][j];          ans = 0;          dfs(1, k);          cout<< ans << endl;      }      system("pause");      return 0;  }  4.2上机实验  4.2.1算法测试   1. 测试数据   3 2  #.#  ###  .#.  4.2.2测试结果及其分析  对应测试数据，其程序执行过程中中间变量的值如图2所示。  程序执行结果如图3所示。  结果分析：  测试数据所创建的二维数组如下图表所示：   |  |  |  | | --- | --- | --- | | **#** | **.** | **#** | | **#** | **#** | **#** | | **.** | **#** | **.** |   测试数据所要求寻找的是不同行不同列的“#”共有多少对，从上表可以看出，符合测试所设条件的“#”共有8组，其结果与程序运行结果一致，说明针对测试数据其程序和算法正确。    图2 程序执行过程中间结果    图3 测试数据对应的程序执行结果  5.实验总结  通过本次实验，在给定算法的情况下，可以初步正确理解算法的执行过程，可以了解算法执行过程中每个变量的变化过程，并使用流程图表示出算法执行情况及每个变量的变化情况。 |