深度优先搜索算法

1. 实验题目

利用深度优先搜索算法（DFS）搜索goal

1. 算法简介

深度优先搜索DFS（Depth First Search.）简而言之就是沿着一个节点的一条支路一直搜索到底，看是否能搜到goal，如果不能搜到，则返回目前搜索的节点的上一个节点，搜索它的另外一条支路，知道找到goal。

DFS的原理与栈相同，先进后出，回溯时先回溯最晚搜到的节点，再回溯最开始搜到的子节点。

1. 实验目的

成功编写一个深度优先算法，将1-4，几个数字进行全排列，列出所有的可能和个数。

1. 实验代码

**package** dfs;

**public** **class** DFS {

**static** **int**[] *arr* = **new** **int**[5];//存储目标数组

**static** **int**[] *result* = **new** **int**[5];

**static** **int** *step\_total* = 5;

**static** **int** *count* = 0;

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

*full*(1);

System.***out***.println("total count:" + *count*);;

}

**public** **static** **void** full(**int** step) {

// **TODO** Auto-generated method stub

**if**(step >= *step\_total*) {

System.***out***.println(*arr*[1] + "" + *arr*[2]+ "" + *arr*[3] + "" + *arr*[4]);

*count*++;

**return**;

}

//重复内容

**for**(**int** i = 1;i < *step\_total*;i++) {

**if**(*result*[i] == 0) {//如果当前数没有被排列

*result*[i] = 1;//标记当前数已经被排列

*arr*[step] = i;

*full*(step + 1);

*result*[i] = 0;

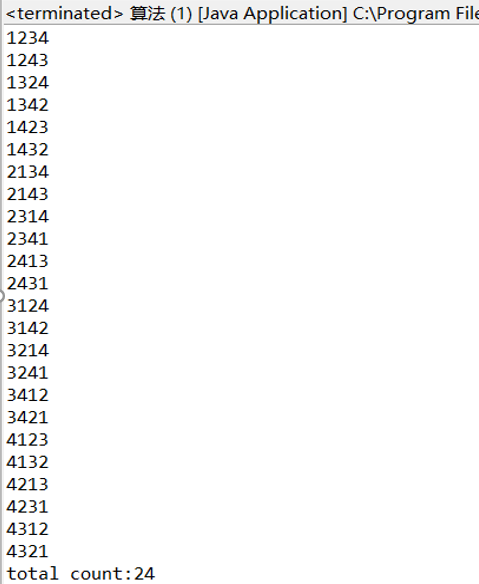
}

}

}

}

1. 实验结果



1. 实验总结

DFS的空间复杂度没有BFS要求的那么多，是可以实现的，但是唯一的缺点是，不具备完整性，因为可能陷入某个支点无限循环导致不能找到解，所以也不具备最优性，但实现复杂度与广度优先一致，所以还是可以运行的。