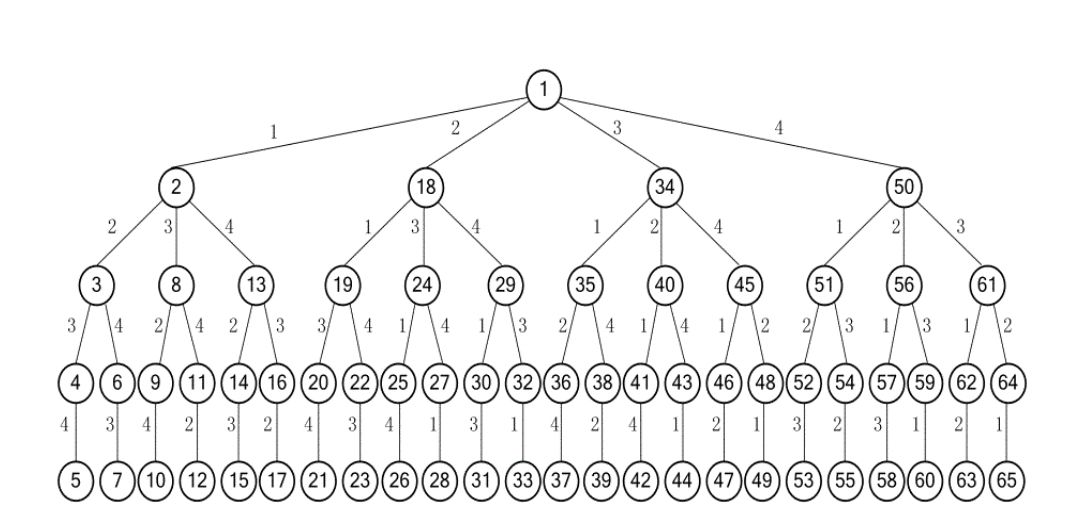
1 ．**4\*4皇后问题**

**状态空间：**

**解空间：16\*15\*14\*13=32760**

**解空间树：**

****

**采用深度优先算法，可以求出4\*4皇后问题可行解有2个，如图：**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **1** |  |  |
|  |  |  | **4** |
| **2** |  |  |  |
|  |  | **3** |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **1** |  |
| **2** |  |  |  |
|  |  |  | **4** |
|  | **3** |  |  |

**用栈，以及回溯法来做**

**#include<stdio.h>**

**#include<math.h>**

**int place(int b);**

**int a[10]; //定义栈来记录皇后位置**

**int main(void)**

**{**

**int n; //输入**

**int k; //第k个皇后**

**int j; //输出结果用**

**scanf("%d",&n);**

**a[1]=0;**

**k=1; //第一个皇后开始**

**while(k>0)**

**{**

**a[k]=a[k]+1; //第一次是从1开始,以后是退回前一个皇后时这个皇后的位置向后挪一位**

**while(a[k]<=n&&!place(k)) // 位置合法？**

**a[k]=a[k]+1; //不合法再向后挪一位**

**if(a[k]<=n) //位置没有超过合法位置**

**{**

**if(k==n) //全部皇后安放完**

**{**

**for(j=1;j<=n;j++)// 输出位置**

**printf("%3d",a[j]);**

**printf("\n");**

**}**

**else //还有皇后没有放**

**{**

**k++; //继续放下一个皇后**

**a[k]=0; //初始化位置**

**}**

**}**

**else //位置不合法**

**k--; //退回去，回溯**

**}**

**return 0;**

**2．分析深度优先搜索4个特性：**

**①完整性（completeness）：**

**完整性是指能否判定问题是否有解，并且当一个问题存在解时，能否把问题的所有解都找出来。如果是有限状态的，且没有死循环的，是具备完整性的。但如果是无限状态的，或者具有死循环的，是不具备完整性的。**

**②最优性（optimality）：**

**最优性是指能否判定问题是否有最优解，并且当一个问题存在最优解时，能否把问题的最优解找出来。**

**③时间复杂度（time complexity）：**

**④空间复杂度（space complexity）：**