

小马加编信息学教案(二十)

深度优先搜索

- 一. 课程内容
- 二. 知识讲解
 - 1. 深度优先搜索概念
 - 2. 深度优先搜索实现
 - 3. 深度优先搜索剪枝
- 三. 经典例题
- 四. 提高巩固

一. 课程内容

1. 深度优先搜索简介
2. 深度优先搜索实现
3. 深度优先搜索剪枝

二. 知识讲解

1. 深度优先搜索概念

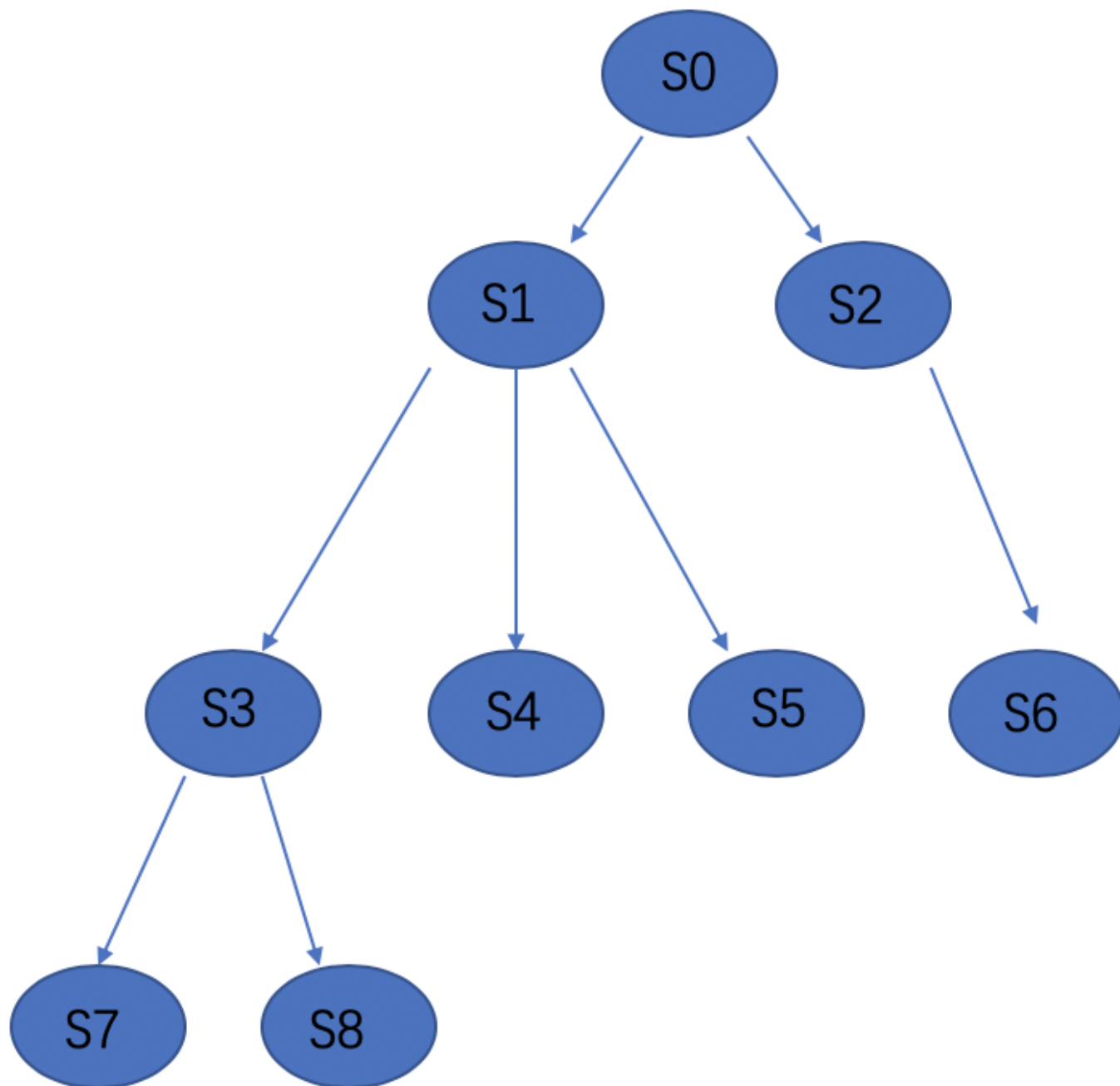
深度优先搜索称为dfs (depth first search)

我们常常用来解决这样的一类问题：

我们需要作出多步决策，每一步决策对后续和结果的影响不容易计算，那么可以使用dfs来枚举所有情况

除此之外，使用dfs枚举所有可能的情况的方法，几乎可以用较低的效率解决所有涉及决策的问题。这就是常说的暴力方法，用以获得部分分。

dfs的方式是在当前状态下，枚举所有可行的决策，依次作出枚举到的决策，然后进入新状态继续枚举。如果没有可选的决策，就退回上一个状态，枚举下一个可选的决策。



我们记S0,S1,...为状态，箭头为做了决策后的状态转移，S0为初始状态。那么深度优先搜索的搜索顺序是由S0->S1->S3->S7，然后退回S3，再转移到S8。再退回S1，再到S4.....（结合例题讲解状态、转移等概念）

注意状态序列，先搜索的状态后退出，后搜索的状态先退出，符合栈先进后出的特点。系统也是使用栈来存储递归结构的。

注意状态之间可能会出现环形转移，即S1能转移到S2，S2也能转移到S1。

由于我们只需要搜索出所有可能出现的状态，那么转移到出现过的状态S'时我们就没有必要继续搜索S'。这样不会出现死循环的情况，并且能搜索到所有状态。

2. 深度优先搜索实现

伪代码

```
void dfs(当前状态s)
{
    if(s为终止状态)
    {
        更新答案 (如评估优劣, 方案计数等)
        return;
    }
    for(每一种转移方式)
    {
        生成新状态s'
        if(s'是没有生成过的状态) dfs(s');
    }
}
```

结合例题讲解

3. 深度优先搜索剪枝

试想我们搜索到一个状态时, 我们已经可以确定接下来无论怎么决策, 都不会对答案产生影响了, 那么这个状态就没有继续搜索的必要了。

比如在最优解问题中, 我们当前状态一定会比已经搜索得出的当前最优解要劣, 那就可以返回了。这个判断、返回的过程我们称为剪枝。

剪枝的角度多种多样, 需要在题目的练习中理解和领悟。

剪枝是dfs常见的优化技巧, 剪枝得当可以有效改善dfs的效率。

三. 经典例题

1. 生成排列

在 $1, \dots, n$ 中选出 r 个数进行排列。按字典序输出所有排列。

$0 < r < n < 20$

输入格式: 第一行两个整数 n, r

输出格式: 每行一个排列, 按字典序升序输出

样例输入

样例输出

样例输入	样例输出
3 2	1 2 1 3 2 1 2 3 3 1 3 2

2. 数的拆分

输入n，输出将n拆分成若干正整数的和的所有方案。按字典序升序输出。
 $n \leq 20$

输入格式： 第一行一个数字n

输出格式： 每行一个拆分方案，按字典序升序输出

样例输入	样例输出
4	1+1+1+1 1+1+2 1+3 2+2 4

3.密码问题

一个有效的密码由L个小写英文字母组成，至少有一个元音和两个辅音。
并且密码中的字母应按照字母表升序出现（如abc有效，而bac无效）
给出C个字母，按字典序升序输出所有由这C个字母组成的长度为L的有效密码。

输入格式：
第一行两个整数L， C
第二行C个不同的小写字母，两个字母间有一个空格
 $3 \leq L \leq 15$
 $1 \leq C \leq 26$

输出格式：
每行一个有效密码，按字典序升序输出

样例输入	样例输出
------	------

样例输入	样例输出
<pre>5 3 a e x y z</pre>	<pre>axy axz ayz exy exz eyz</pre>

四. 提高巩固

1.n皇后问题

在 $n \times n$ 的国际象棋棋盘放 n 个皇后，使它们无法互相攻击，输出每种方案。

皇后攻击范围为它所在的行、列、对角线。

对于一种方案，易知 n 个皇后分别在第 $1 \sim n$ 行。

每种方案输出 n 个数字，表示第1行,第2行,...,第 n 行的皇后所在的列号。

按字典序升序输出。

$n \leq 8$

输入格式: 一个数 n

输出格式: 每行表示一种方案。按字典序升序输出。

样例输入	样例输出
<pre>4</pre>	<pre>2 4 1 3 3 1 4 2</pre>

2.分书问题

有 n 本书， n 个人。每个人都有自己喜欢的一些书。现在要将每本书分给不同的人，使得每个人都喜欢自己得到的书。问总共有多少种方案？

$n \leq 20$

输入格式:

第一行一个数 n

后一个 $n \times n$ 的01矩阵

矩阵中第 i 行第 j 列为1，表示第 i 个人喜欢第 j 本书，为0表示不喜欢。

输出格式:一个数字表示方案数

样例输入	样例输出
------	------

样例输入	样例输出
5 00110 11001 01100 00010 01001	1