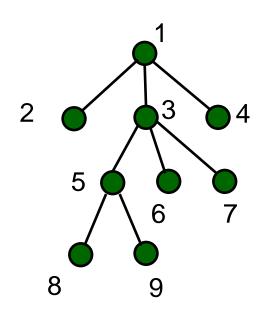
回溯算法的设计思想和适用条件

问题分析

问 题	解 性质	解描述 向量	搜索 空间	搜索 方式	约束 条件
n 后	可行解	$\langle x_1, x_2,, x_n \rangle$ x_i : 第 i 行列号	n叉 树	深度,宽度优先	彼此不 攻击
0-1 背	最优 解	$\langle x_1, x_2,, x_n \rangle$ $x_i = 0,1,$ $x_i = 1 \Leftrightarrow 选 i$	子集树	深度,宽度优先	不超背包 重量
货郎	最优 解	< <i>i</i> ₁ =1, <i>i</i> ₂ ,, <i>i</i> _n > 1,2,,n的排列	排列树	深度,宽度优先	选没有经 过的城市
特点	搜索 解	向量,不断扩 张部分向量	树	跳跃式 遍历	约束条件 回溯判定

深度与宽度优先搜索



深度优先访问顺序:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 8 \rightarrow 9$$
$$\rightarrow 6 \rightarrow 7 \rightarrow 4$$

宽度优先访问顺序:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$$
$$\rightarrow 7 \rightarrow 8 \rightarrow 9$$

回溯算法基本思想

- (1) 适用: 求解搜索问题和优化问题.
- (2) 搜索空间:树,结点对应部分解向量,可行解在树叶上.
- (3) 搜索过程:采用系统的方法隐含 遍历搜索树.
- (4) <mark>搜索策略</mark>:深度优先,宽度优先, 函数优先,宽深结合等.

回溯算法基本思想(续)

- (5) 结点分支判定条件: 满足约束条件---分支扩张解向量 不满足约束条件,回溯到该结点的 父结点.
- (6) 结点状态: 动态生成 白结点(尚未访问) 灰结点(正在访问该结点为根的子树) 黑结点(该结点为根的子树遍历完成)
- (7) 存储: 当前路径

结点状态

深度优先

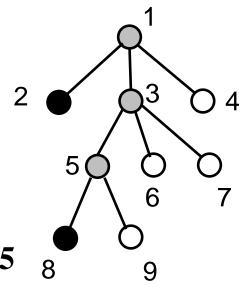
访问次序:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 8$$

已完成访问: 2,8

已访问但未结束: 1,3,5

尚未访问: 9, 6, 7, 4



回溯算法的适用条件

在结点 $\langle x_1, x_2, ..., x_k \rangle$ 处 $P(x_1, x_2, ..., x_k)$ 为真 \Leftrightarrow 向量 $\langle x_1, x_2, ..., x_k \rangle$ 满足某个性质 (n后中 k个皇后放在彼此不攻击的位置)

多米诺性质

$$P(x_1,x_2,...,x_{k+1}) \rightarrow P(x_1,x_2,...,x_k) \quad 0 < k < n$$

 $\neg P(x_1, x_2, ..., x_k) \rightarrow \neg P(x_1, x_2, ..., x_{k+1})$ 0<k < n k 维向量不满足约束条件,扩张向量到 k+1维仍旧不满足,可以回溯.

一个反例

例 求不等式的整数解

$$5x_1+4x_2-x_3\leq 10$$
, $1\leq x_k\leq 3$, $k=1,2,3$ $P(x_1,\ldots,x_k):$ 将 x_1,x_2,\ldots,x_k 代入原不等式的相应部分,部分和小于等于10

不满足多米诺性质:

$$5x_1 + 4x_2 - x_3 \le 10 \implies 5x_1 + 4x_2 \le 10$$

变换使得问题满足多米诺性质:

小结

- 回溯算法的适用条件: 多米诺性质
- 回溯算法的设计步骤
 - (1) 定义解向量和每个分量的取值范围 解向量为 $< x_1, x_2, ..., x_n >$ 确定 x_i 的取值集合为 X_i , i = 1, 2, ..., n.

小结(续)

- (2) 在 $\langle x_1, x_2, ..., x_{k-1} \rangle$ 确定如何计算 x_k 取值集合 S_k , $S_k \subseteq X_k$
- (3) 确定结点儿子的排列规则
- (4) 判断是否满足多米诺性质
- (5) 确定每个结点分支的约束条件
- (6) 确定搜索策略: 深度优先,宽度优先等
- (7) 确定存储搜索路径的数据结构