几个回溯算法的例子

4后问题

4后问题:在4×4的方格棋盘上放置4个皇后,使得没有两个皇后在同一行、同一列、也不在同一条45度的斜线上.问有多少种可能的布局?

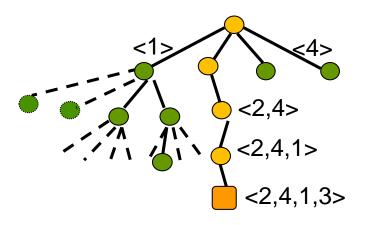
解是 4 维向量 < x₁, x₂, x₃, x₄ > 解: <2,4,1,3>, <3,1,4,2>

推广到8后问题

解: 8维向量,有92个.

例如: <1,5,8,6,3,7,2,4>是解.

搜索空间: 4叉树



每个结点有4个儿子,分别代表选择 1,2,3,4列位置 第 *i* 层选择解向量中第 *i* 个分量的值 最深层的树叶是解 按深度优先次序遍历树,找到所有解

0-1背包问题

问题:

有n种物品,每种物品只有 1个. 第i 种物品价值为 v_i ,重量为 w_i ,i=1,2,...,n. 问如何选择放入背包的物品,使得总重量不超过 B,而价值达到最大?

实例:

 $V=\{12,11,9,8\}, W=\{8,6,4,3\}, B=13$

最优解:

<0,1,1,1>,价值:28,重量:13

算法设计

解: n维0-1向量 $\langle x_1, x_2, ..., x_n \rangle$, $x_i = 1 \Leftrightarrow$ 物品 i 选入背包

结点: $\langle x_1, x_2, ..., x_k \rangle$ (部分向量)

搜索空间: 0-1取值的二叉树, 称为子集树,有 2^n 片树叶.

可行解:满足约束条件 $\sum_{i=1}^{n} w_i x_i \leq B$ 的解最优解:可行解中价值达到最大的解

实例

输入:

 $V=\{12,11,9,8\}, W=\{8,6,4,3\}, B=13$

2个可行解:

<0,1,1,1>, 选入物品2,3,4,价值为28,

重量为13

<1,0,1,0>, 选入物品1,3,价值为21,

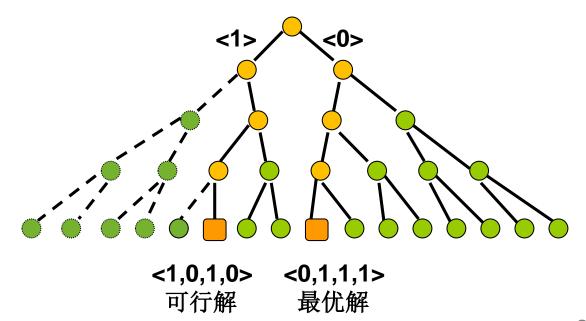
重量为12

最优解: <0,1,1,1>

搜索空间

实例:V={12,11,9,8}, W={8,6,4,3}, B=13

搜索空间:子集树,2ⁿ片树叶



货郎问题

问题: 有n个城市,已知任两个城市 之间的距离,求一条每个城市恰好经 过一次的回路,使得总长度最小.

建模: 城市集 $C = \{c_1, c_2, ..., c_n\}$, 距离 $d(c_i, c_i) = d(c_i, c_i) \in \mathbb{Z}^+$, $1 \le i < j \le n$

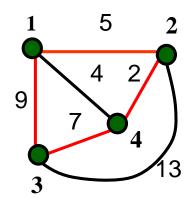
求: 1,2,...,n的排列 $k_1,k_2,...,k_n$ 使得

$$\min\{\sum_{i=1}^{n-1} d(c_{k_i}, c_{k_{i+1}}) + d(c_{k_n}, c_{k_1})\}$$

实例

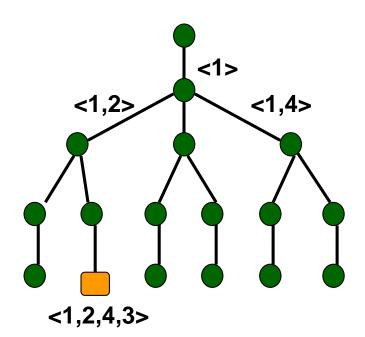
$$C = \{1,2,3,4\}$$

 $d(1,2)=5, d(1,3)=9,$
 $d(1,4)=4, d(2,3)=13,$
 $d(2,4)=2, d(3,4)=7$



搜索空间

排列树, 有(n-1)!片树叶



小结

- 回朔算法的例子: n后问题, 0-1背 包问题, 货郎问题
- 解: 向量
- 搜索空间: 树,可能是n叉树、子 集树、排列树等等,树的结点对应 于部分向量,可行解在叶结点
- 搜索方法: 深度优先, 宽度优先, ... 跳越式遍历搜索树, 找到解