回溯算法的实现及实例

回溯算法递归实现

算法 **ReBack** (k)

1. if $k>n$ then $<x_1,x_2,\ldots,x_n>$是解
2. else while $S_k \neq \varnothing$ do
3. $x_k \leftarrow S_k$ 中最小值
4. $S_k \leftarrow S_k - \{ x_k \}$
5. 计算 S_{k+1}
6. **ReBack** (k+1)

算法 ReBacktrack (n)

输入：n

输出：所有的解

1. for $k \leftarrow 1$ to n 计算 X_k且$S_k \leftarrow X_k$
2. ReBack (1)

迭代实现

迭代算法 Backtrack

输入：n

输出：所有的解

确定初始取值

1. 对于 i =1, 2, … , n 确定X_i
2. $k \leftarrow 1$
3. 计算 S_k

满足约束分支搜索

4. while $S_k \neq \varnothing$ do
5. $\quad x_k \leftarrow S_k$中最小值；$S_k \leftarrow S_k - \{x_k\}$
6. $\quad$ if $k < n$ then
7. $\quad\quad k \leftarrow k+1$；计算 S_k
8. $\quad$ else $< x_1, x_2, \ldots, x_n >$是解
9. if $k > 1$ then $k \leftarrow k-1$；goto 4

回溯

装载问题

问题：有n个集装箱，需要装上两艘载重分别为c_1和c_2的轮船. w_i为第 i 个集装箱的重量，且 $w_1+w_2+\ldots+w_n \leq c_1+c_2$

问：是否存在一种合理的装载方案把这 n 个集装箱装上船？如果有，请给出一种方案.

实例：

$W = < \underline{90},80,\underline{40},30,20,\underline{12,10} >$

c_1=152，c_2=130

解: 1,3,6,7装第一艘船，其余第2艘船

求解思路

输入: $W=<w_1,w_2, ..., w_n>$为集装箱重量
c_1和c_2为船的最大载重量

算法思想：令第一艘船的装入量为W_1,

1. 用回溯算法求使得c_1-W_1达到最小的装载方案.
2. 若满足

$$w_1+w_2+...+w_n-W_1\leq c_2$$

则回答“Yes”，否则回答“No”

伪码

B为当前空隙
best 最小空隙

算法 Loading (W, c_1),

1. Sort(W);
2. $B \leftarrow c_1$; $best \leftarrow c_1$; $i \leftarrow 1$;
3. while $i \leq n$ do
4. if 装入 i后重量不超过 c_1
5. then $B \leftarrow B - w_i$; $x[i] \leftarrow 1$; $i \leftarrow i+1$;
6. else $x[i] \leftarrow 0$; $i \leftarrow i+1$;
7. if $B < best$ then 记录解； $best \leftarrow B$;
8. Backtrack(i)； 回溯
9. if $i=1$ then return 最优解
10. else goto 3 .

子过程 Backtrack

算法 Backtrack(i)

1. **while** $i>1$ **and** $x[\,i\,]=0$ **do**
2. $\quad i \leftarrow i-1$;
3. **if** $x[i]=1$
4. **then** $x[i] \leftarrow 0$;
5. $\qquad B \leftarrow B+w_i$;
6. $\qquad i \leftarrow i+1$.

沿右分支一
直回溯发现
左分支边，
或到根为止

实例

$W = < 90,80,40,30,20,12,10 >$

c_1=152，c_2=130

解：

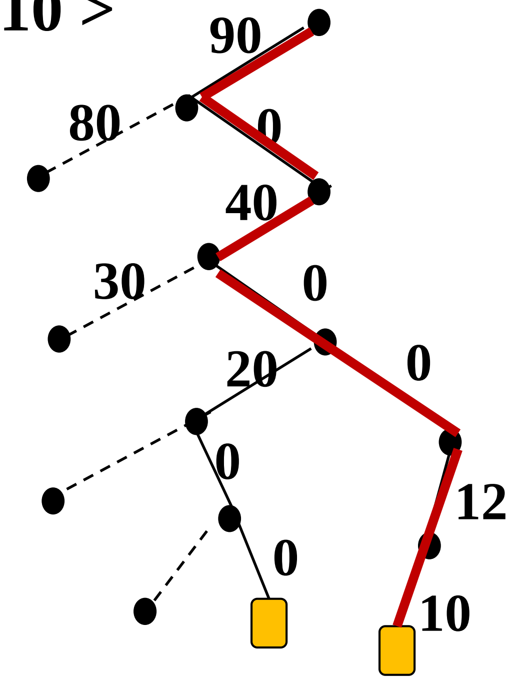
可以装，方案如下：

1,3,6,7 装第一艘船

2,4,5 装第二艘船

时间复杂性：

$W(n)=O(2^n)$



小结

- 回溯算法的实现：
  递归实现、迭代实现

- 装载问题
  问题描述
  算法伪码
  最坏情况下时间复杂度$O(2^n)$