回溯子集树与排列树——装载问题&旅行售货员问题（算法设计课题）

2018.07.19 16:31 156 浏览

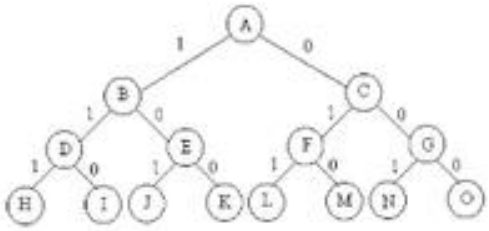
*对回溯法不是很理解的请移步博客 http://blog.csdn.net/sm9sun/article/details/53244484

掌握了回溯法以后，我们给出两种定义：

当所给问题是从 n 个元素的集合 S 中找出满足某种性质的子集时，解空间为子集树。例如：0-1 背包问题

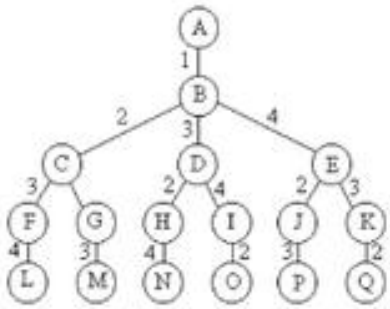
当所给问题是从 n 个元素的集合 S 中找出满足某种性质的排列时，解空间为排列树。例如：旅行售货员问题

子集树与排列树



遍历子集树需$O(2^n)$计算时间

```
void backtrack (int t)
{
  if (t>n) output(x);
    else
      for (int i=0;i<=1;i++) {
        x[t]=i;
        if (legal(t)) backtrack(t+1);
      }
}
```



遍历排列树需要$O(n!)$计算时间

```
void backtrack (int t)
{
  if (t>n) output(x);
    else
      for (int i=t;i<=n;i++) {
        swap(x[t], x[i]);
        if (legal(t)) backtrack(t+1);
        swap(x[t], x[i]);
      }
}
```

代码：

子集树

```
void backtrack (int t){if (t>n) output(x);elsefor (int i=0;i<=1;i++) {x[t]=i;if (legal(t)) backtrack(t+1);}}
```

排列树

```
void backtrack (int t){if (t>n) output(x);elsefor (int i=t;i<=n;i++) {swap(x[t], x[i]);if (legal(t)) backtrack(t+1);swap(x[t], x[i]);}}
```

题目描述：

有一批共 n 个集装箱要装上 2 艘载重量分别为 c1 和 c2 的轮船，其中集装箱 i 的重量为 wi，且，装载问题要求确定是否有一个合理的装载方案可将这些集装箱装上这 2 艘轮船。如果有，找出一种装载方案。

例如：当 n=3,c1=c2=50,且 w=[10,40,40]时，则可以将集装箱 1 和 2 装到第一艘轮船上，而将集装箱 3 装到第二艘轮船上；如果 w=[20,40,40]，则无法将这 3 个集装箱都装上轮船。

解题思路： 容易证明，如果一个给定装载问题有解，则采用下面的策略可得到最优装载方案。

(1)首先将第一艘轮船尽可能装满；

(2)将剩余的集装箱装上第二艘轮船。

将第一艘轮船尽可能装满等价于选取全体集装箱的一个子集，使该子集中集装箱重量之和最接近 C1。由此可知，装载问题等价于以下特殊的 0-1 背包问题。

```
/************************************************************** *问 题：装载问题 *算 法：回溯法 *描 述：解空间为 子集树 **************************************************************/

#include<stdio.h>

int x[1001],w[1001];

int cw,n,c,r,bestw;

void backtrack(int i){

if (i>n){

bestw=cw;
```

```
return;

}

r-=w[i];

if (cw+w[i]<=c){

x[i]=1;

cw+=w[i];

backtrack(i+1);

cw-=w[i];

}i

f (cw+r>bestw){

x[i]=0;

backtrack(i+1);

}

r+=w[i];

}

int main(){

int i;

while(~scanf("%d",&n))

{

for(i=1;i<=n;i++){

scanf("%d",&w[i]);
```

```
x[i]=0;

r+=w[i];

}

scanf("%d",&c);

cw=bestw=0;

backtrack(1);

printf("%d\n",bestw);}}
```

题目描述：

某售货员要到若干城市去推销商品，已知各城市之间的路程（旅费），他要选定一条从驻地出发，经过每个城市一遍，最后回到驻地的路线，使总的路程（总旅费）最小。

解题思路：

旅行售货员问题的解空间是一颗排序树，对于排序树的回溯法搜索与生成 1，2，3，4，…，n 的所有排列的递归算法 Perm 类似，开始时，x = [1,2,…,n],则相应的排序树由 x[1:n]的所有排序构成。以下解释排序算法 Perm：
假设 Perm(1)的含义是对 x = [1,2,…,n]进行排序，则 Perm(i)的含义是对[i,i+1,i+2,…,n]进行排序。为了方便描述，规定一个操作 P(j),表示在数组 x 中，第 j 个与第一个交换，于是得到递归关系式： Perm(i) = P(i)Perm(i+1) + P(i+1)Perm(i+1) + …+P(n)Perm(i+1)，旅行售货员的回溯法 Backtrack 在 Perm 的基础上使用了剪枝函数，将无效的全排列和有效的全排列非最优的次序统统都舍去。