# 选择知识点

## 回溯法

把问题的解空间转化成了图或者树的结构表示，然后使用**深度优先搜索**策略进行遍历，遍历的过程中记录和寻找所有可行解或者最优解。

## 多项式方案

多项式级时间： n出现在底数的位置

非多项式级时间： 计算复杂度太高

一个多项式近似方案(PAS)是一个近似方案,其中每一个算法在输入实例I的规模的多项式时间内运行;

一个完全多项式近似方案(FPAS)是一个近似方案,其中每个算法在以输入实例的规模I和两者的多项式时间内运行.

若P2多项式时间转化为(polynomial transforms to) P1,则P2至少不会比P1难。P2比P1简单，所以才能用P1的表示。

多项式算法：时间复杂度是输入**数据量**的多项式表达

伪多项式算法：时间复杂度是输入**数值**的多项式表达

## NP

P问题：可以找到一个能在多项式的时间里解决它的算法。

NP问题：可以在多项式的时间里验证一个解是否正确的问题。(并不是求出解，而是猜测一个解，验证其是否正确)

NPC(完全)问题：

属于一个NP问题。

所有NP问题都可以约化成该问题。

## 算法正确性

算法正确性是对任意一个合法的输入经过有限步执行之后算法应给出正确的结果。

## 常见算法介绍

Las Vegas(拉斯维加斯)：

一种随机搜索解的算法，不一定能找到解，但找到的一定是解。

Monte Carlo 算法

一种根据概率统计设计的随机算法，是一种近似算法。一定有解，但解不一定正确。

贪心算法：

选择都是可行的、局部最优的，但不一定是全局最优。

## 近似算法

近似算法解：c

最优解：

性能比：

## 各种排序的时间复杂度

：不大于，最坏情况

：不小于，最好情况

快排：

# 简答知识点

## 减治策略

利用了一个问题给定实例的解 和 同样问题较小实例的解 之间的关系。

1. **减一个常量**：常常是规模减1，如：插入排序
2. **减一个常因子：**每次迭代减去一个常量的倍数(小于1)，如：折半查找。
3. **减可变规模**：每次都减小，但减小的程度不是确定的，如：二叉查找树.

## 查找树

### 二叉查找树

最优情况：满二叉树，高度相同，

最差情况：单二叉树，每层只有一个节点，

插入数据有序排列时，二叉树变为一条长链。

为了防止出现最差情况，可以选择AVL树和2-3树进行查找和插入

### AVL树

AVL树是自平衡二叉查找树，任一节点对应的两棵子树的最大高度差为1，因此它也被称为高度平衡树。

查找、插入和删除在平均和最坏情况下的时间复杂度都是

### 2-3树

一种平衡树，各种情况的时间复杂度都是。

2节点：有1个数据，2个子节点

3节点：有2个数据，3个子节点

由2节点、3节点和叶节点构成的树叫做2-3树

## Monte Carlo算法转化为Las Vegas 算法

Monte Carlo算法得到结果后进行正确性验证，正确的情况下才返回解，否则不返回。

## 0-1背包问题

有n个物体，重量和价值分别为，求一个承重为W的背包中最多能装有多少价值的物体。

多项式等价的判定问题：在上述条件下，一个承重为W的背包中装有的物体价值是否？

等价原因：若判定问题存在多项式时间的解法，那么有限次内调用该解法，即可解决0-1背包问题。

# 分治

## 逆序数

写出一个求解下列问题的分治算法,推导其时间复杂性并与蛮力法相比较。 给定互不相等的n个数的一个序列若其中某两个数和的关系为: 且𝑖 < 𝑗,则称和是逆序的。要求计算该序列中的逆序个数,即具有逆序关系的元素对的总数目.

解：

总体思路：当待统计数组个数大于1时，将数组分为两部分分别统计逆序数，然后在统计这两个子数组之间的逆序数，并且在统计时对两个子数组进行归并排序。总逆序数等于这三者的和。

count(A, begin\_i, end\_i) ：递归统计数组A[begin\_i,end\_i]内的逆序数

combine\_count(A, l\_begin\_i, l\_end\_i, r\_begin\_i, r\_end\_i) :

统计数组A[l\_begin\_i,l\_end\_i]与数组A[r\_begin\_i,r\_end\_i]之间的逆序数

伪码：

count(A, begin\_i, end\_i){

if(begin\_i < end\_i) then{

temp\_i = (begin\_i + end\_i)/2 #取整

count(A, begin\_i, temp\_i)

count(A, temp\_i, end\_i)

combine\_count(A, begin\_i, temp\_i, temp\_i, end\_i)

}

}

combine\_count(A, begin\_i, temp\_i, temp\_i, end\_i){

l\_A = A[begin\_i, temp\_i]

r\_A = A[temp\_i, end\_i]

i = j = 1

k = begin\_i

while(i <= l\_A.length && j <= r\_A.length) do{

if l\_A[i] <= r\_A[j] then

A[k] = l\_A[i]

i ++

k ++

else

A[k] = r\_A[j]

j ++

k ++

count\_num += l\_A.length + j - i

}

}

count\_num = 0 #计数器，全局变量

main(A){

count(A, 1, A.length)

}

分析：

暴力求解法：

分治法：

每层的时间复杂度为，有层

总时间复杂度为

# 变治

## 求多数元素

方法和3相同，先归并排序，时间复杂度为，然后再遍历一遍统计元素出现次数，时间复杂度为。最终时间复杂度为

# 动态规划

有8万元的投资可以投给3个项目，每个项目在不同投资数额下（以万元为单位）的利润如下表：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 项目1 | 0 | 5 | 15 | 40 | 80 | 90 | 95 | 98 | 100 |
| 项目2 | 0 | 5 | 15 | 40 | 60 | 70 | 73 | 74 | 75 |
| 项目3 | 0 | 4 | 26 | 40 | 45 | 50 | 51 | 52 | 53 |

请安排投资计划，使总的利润最大。

写出你所设的状态变量、决策变量、状态转移方程与递推关系式，和手工求解的详细步骤和结果。

假设变量：

：在前k个项目中投资x万元可以获得的最大利润,

：在第i个项目上投资j万元能够获得的利润

状态变量：前k个项目的总投资金额

决策变量：第k个项目的投资金额，

状态转移方程：

递推关系式：

k=1时：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| k x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 0 | 5 | 15 | 40 | 80 | 90 | 95 | 98 | 100 |

k=2时：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| k x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 0 | 5 | 15 | 40 | 80 | 90 | 95 | 98 | 100 |
| 2 | 0 | 5 | 15 | 40 | 80 | 90 | 95 | 120 | 140 |

k=2时：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| k x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 0 | 5 | 15 | 40 | 80 | 90 | 95 | 98 | 100 |
| 2 | 0 | 5 | 15 | 40 | 80 | 90 | 95 | 120 | 140 |
| 3 | 0 | 5 | 26 | 40 | 80 | 90 | 106 | 120 | 140 |

最优140，

某工厂调查了解市场情况，估计在今后4个月内，市场对其产品的需求量如下表所示：

|  |  |
| --- | --- |
| 时期(月) | 需求量(产品单位) |
| 1 | 2 |
| 2 | 3 |
| 3 | 2 |
| 4 | 4 |

已知：

对每个月来讲，生产一批产品的固定成本费为3(千元)，若不生产则为零。

每生产单位产品的成本费为1(千元)。

同时，在任何一个月内，最大生产量不超过6(千元).

每单位产品的库存费用为每月0.5(千元)

同时要求第一月初和第四个月末，均无产品库存。

问：在满足上述条件下，该厂应该如何安排各个时期的生产使所花费的总成本最低？

写出你所设的状态变量、决策变量、状态转移方程与递推关系式，和手工求解的详细步骤和结果。

# 分支定界法

## AB国贸易问题

### 构造二叉树的方式

将城市作为二叉树的节点，其中根节点为城市A，

对于任意一个节点，都求得一个活节点集合，中的任意一元素满足以下条件：

1. 与对应的城市直接相连
2. 不是的某个祖先节点，即不在从根节点到的简单路径中。

从根节点开始，求出该节点对应的活节点集合，以这些活节点作为子节点。然后以某种方法从这些子节点中选取一个作为下一个节点，重复上述操作。

### 原则

前进：

1. 当前节点不是城市B，
2. 不为空
3. 从根节点到当前节点花费时间小于T

分支：

当前节点不是城市B，且不为空时，以中所有元素作为子节点，并从中选择一个作为新节点进行遍历。选择条件如下：

1. 如果B在中，优先选择B
2. 否则，优先选择中与B相连的且税费最低的
3. 否则，优先选择中与C21~C50相连的且税费最低的
4. 否则，优先选择税费最低的

回溯：

1. 当前节点是城市B
2. 或为空
3. 或从根节点到当前节点花费时间大于等于T
4. 或从根节点到当前节点的简单路径所导致的税费已经不小于已求得的最小解

剪枝：

1. 当前节点不是B，且从根节点到当前节点的简单路径所导致的税费已经不小于已求得的最小解，则进行剪枝。
2. 当前节点不是B，且从根节点到当前节点花费的时间已经不小于T，则进行剪枝。