NANJING UNIVERSITY

算法设计与分析基础《Introduction to the Design and Analysis of Algorithms》

蛮力法

南京大学软件学院

李传艺

lcy@nju.edu.cn

费彝民楼917

目录

- 蛮力法回顾
  - 概念
  - 选择排序
  - 插入排序
- 顺序查找
- 蛮力字符串匹配
- 最近对问题
- 凸包问题
- 穷举查找

蛮力法回顾

- 枚举法、穷举法、暴力解法
  - 在不考虑时间、空间效率的情况下，寻求问题的解决方案
- 体现蛮力法的一些解决问题的方案
  - 搜索所有解空间
    - 找约束条件、找枚举范围：在搜索前尽可能减小搜索空间
  - 搜索所有路径
  - 直接计算
  - 模拟和仿真

- 选择排序
  - 遍历列表，找到最大或最小的与第一个没有排序好的交换
  - 时间复杂度
    - $\Theta(n^2)$

- 冒泡排序
  - 遍历列表，每次都把与当前元素与下一个是逆序的元素交换，把最大或最小的元素排到最后
  - 时间复杂度
    - $\Theta(n^2)$

蛮力之顺序查找

- 如何在给定的列表中查找一个给定的值
  - 遍历列表直到找到给定的元素
  - 需要判断列表是否结束
    - 技巧：将给定元素放到列表结尾，最后判断一次是否是列表的结尾
  - 时间复杂度：$\Theta(n)$

- 确定搜索空间，在空间内根据限制条件顺序查找
- 百鸡百钱问题
  - 中国古代数学家张丘建在《算经》中提出了著名的“百鸡百钱问题”：鸡翁一，值钱五；鸡母一值钱三；鸡雏三，值钱一；百钱买百鸡，翁、母、雏各几何？

  - 算法设计1
    - 3个需要确定的值，各有空间；有一个限制条件
    - 遍历整个搜索空间
  - 算法设计2
    - 三个变量之间就有一个限制条件，只要遍历其中两个的空间即可
    - 额外发现雏必须是3的倍数

例子

- 求所有的三位数，它除以11所得的余数等于它的三位数字的平方和。

- 三位数只有900个
- 除以11的余数只有11种可能，0不算，只有10个
- 三个数平方和小于等于10的必定都小于等于3
- 百位数不等于0

- 某公司有N层楼，为了测出哪层楼最高，可以使用一种仪器从天花板向地板自由落体，仪器不会摔坏；如果有两个一模一样的仪器，请设计一个最佳效率类型的算法解决这个问题。
  - 效率如何衡量？比较次数？移动次数？额外空间？

蛮力之字符串匹配

- 问题描述
  - 给定一个n个字符组成的串，称为文本，一个m(m<=n)个字符的串，称为模式，从文本中寻找匹配模式的子串

- 蛮力解法
  - 以文本中的每一个字符为开始字符，用模式串去匹配，直到文本结束
  - 最糟糕的情况下算法效率属于$\Theta(nm)$
  - 可以认为大多数移动的情况都是发生在很少的比较次数之后，因此算法的平均效率比最坏情况要好的多
    - 随机查找文本的时候，显示出线性效率$\Theta(n+m)=\Theta(n)$

- 给出一个蛮力字符串匹配效率最差的例子？
  - 每一次移动尽可能多的比较？

- 例. 给定一个文本CADFBAAXBYA，计算以A开始以B结束的字符串个数
  - 联想一下：区间包含点个数的问题？是否有关联？

蛮力之最近对问题

- 找出一个包含n个点的集合中距离最近的两个点

- 蛮力解法
  - 求出每两个点之间的距离，选出最小的那个
  - 不重复计算两个点之间距离：i=0; j=i+1
  - 不计算具体距离，使用平方和代表距离

  - 复杂度为$\Theta(n^2)$

- 如果是一条线上的n个点如何计算距离最小的两个点？
- 计算距离最近点的算法适用于各种不同的距离计算方法？
  - 欧几里得距离
  - 曼哈顿距离

- 如果是在K维空间中，蛮力最近对问题的算法效率如何？

蛮力之凸包问题

- 凸集合
  - 对于平面上的一个点集合（有限的或者无限的），如果以集合中任意两点P和Q为端点的线段都属于该集合，我们说这个集合是凸的
  - 凸集合例子：直线、三角形和任意凸多边形等
  - 非凸集合例子：五角星、月牙形状等
- 凸包
  - 一个点集合S的凸包是包含S的最小凸集合（“最小”是意指，S的凸包一定是所有包含S的凸集合的子集）
  - 例子
    - 如果S是凸的，它的凸包是它本身
    - 如果S只有两个点、如果S只有三个共线的点、如果S只有三个不共线的点
- 凸包定理
  - 任意包含多于两个不共线点的集合S的凸包肯定是以S中某些点为顶点的凸多边形
- 凸包问题
  - 为一个包含n个点的集合构造凸包的问题

蛮力之凸包问题——极点

- 极点：凸包的顶点
- 找到极点、找到极点连接的顺序
- 蛮力解法？
  - 极点的特性？极点连线的特性？
  - 如果两个点构成的线段是凸包的边界，则其它点都在这条线段所在直线的一边
  - 复杂度$\Theta(n^3)$
- 思考：设计一个线性算法找到n>1个点集合的两个极点

蛮力之凸包问题——极点和线性规划问题

- 线性规划问题
  - 多变量线性函数的最优化问题；这些变量满足一些线性等式或者不等式形式的约束

- 例. 求出3x+5y在下列约束下的最大值：x+y<=4且x+3y<=6且x>=0、y>=0

- 约束条件构成一个可行区域：凸包
- 最优解和凸包的关系
  - 最优解一定出现在凸包的某个极点上

- 线性规划转换为求凸包极点的问题

蛮力之穷举查找

- NP问题的定义——不确定算法
  - 变换为猜测阶段和验证阶段
  - 验证阶段可以使用多项式复杂度解法完成

- 猜测阶段形成的解空间是指数级增长的
  - 一般为组合、排列问题

- 穷举查找：用户在指数级增长的解空间中查找满足要求的解
  - 解决最优问题的蛮力解法

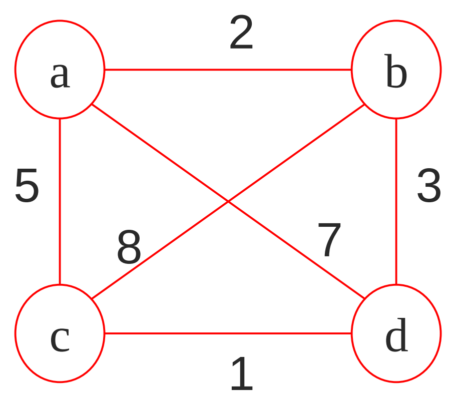
- 问题
  - 旅行商问题
  - 背包问题
    - （以上都是NP完全问题：其他NP问题能够在多项式时间内转化为该问题）
  - 分配问题

旅行商问题（1）

- 旅行商问题（Traveling salesman problem, TSP）
  - 又名“货郎担问题”或“旅行推销员问题”
  - 有一个旅行商从城市1出发，需要到城市2，3，…，n去推销货物，最后返回城市1，若任意两个城市之间的距离已知，则该旅行商应如何选择其最佳行走路线？
  - TSP在图论意义下又被称为最小哈密顿回路问题：对图的每一个点都经过一次最后回到原点的路径问题。
  - 有很多可以抽象为TSP的实际问题，例如：
    - 投送邮件、快递
    - 最省路费、时间的旅游路径规划
    - 印制电路板穿孔问题
    - 多个卫星位置调整
    - 电缆、光缆布线等

旅行商问题（2）

- 具体例子
- 有a、b、c、d四个城市，两两之间的距离如图所示，找到从a出发经过其它所有点后回到a的最短路径



穷举法解：

a→b→c→d→a　　length=2+8+1+7=18

a→b→d→c→a　　length=2+3+1+5=11　　最佳

a→c→b→d→a　　length=5+8+3+7=23

a→c→d→b→a　　length=5+1+3+2=11　　最佳

a→d→b→c→a　　length=7+3+8+5=23

a→d→c→b→a　　length=7+1+8+2=18

优化：定义中间一条路径的方向，因为不同方向的值是相同的，例如只考虑b在c之前的路径

旅行商问题（3）

- 旅行商问题的扩展
  - 瓶颈TSP——简称BTSP
    - 经过的最长距离最短
    - 最小化瓶颈距离
    - 仍然是NP难问题
  - 最小比率TSP——简称MRTSP
    - 除了行程还会产生收益
    - 优化目标：总行程与总收益比最小
    - 与单纯的总行程最短比，更具有实际意义
  - 多人TSP——简称MTSP
    - 多个推销员同时出发，走不同路线，使得所有城市至少被访问一次
    - 所有人的总路程最小

背包问题（1）

- 给定*n*个，重量为*w*1, *w*2, …, *wn*、价值为*v*1,…,*vn*的物品和一个承重为*W*的背包，求这些物品中一个最有价值的子集，并且要能够装到背包中。

- 出现在多种实际应用中
  - 寻找如何削减原材料使得浪费最少
  - 选择投资和投资组合
  - 测试的构建和评分中：如何选择满足要求的题目解答使得得分最高

- 1998年石溪布鲁克大学（Stony Brook University, SBU）算法库的调查统计中
  - 理科非常好，物理前十，纯数学前20，计算机前40，文商科比较水
  - 75个算法问题中最受欢迎的
  - 最需要解决的算法问题中排第四
    - KD树：高维数据查询使用的索引数据结构
    - 后缀树：可用于存储字符串预处理得到的信息，加速字符串匹配等操作
    - 装箱问题(bin packing)：往箱子里装东西，至少需要多少个箱子

背包问题（2）

- 例. 一个大小为10的背包，装入大小和价值为（7，42），（3，12），（4，40）和（5，25）的四个物品，如何装使得装入物品价值最大。
- 穷举查找

物品集合	总重量	总价值
1	7	42
2	3	12
3	4	40
4	5	25
1，2	10	36
1，3	11	
1，4	12	
2，3	7	52
2，4	8	37
3，4	**9**	**65**
1，2，3	14	
1，2，4	15	
1，3，4	16	
2，3，4	12	
1，2，3，4	19	

背包问题（3）

- 包含n个元素的集合的子集的个数为2^n
- 穷举查找方案复杂度肯定满足$\Omega(2^n)$

- 背包问题的扩展
  - 完全背包：每一件物品不计件数
  - 多重背包：每一个物品对应有一个确定的件数

- 以上是穷举查找用于解决NP问题的例子，如果搜索空间不大，可以节约思考的时间；同时注意对搜索空间的缩减优化

例题

- 如何对排序问题使用穷举查找的方法？算法的复杂度如何？

- 考虑划分问题：给定n个整数，把它们划分为元素和相同但不相交的子集（不一定总有解），设计一个穷举查找的算法。

- 完全（完备）子图问题
  - 给定一个图G和正整数k，确定该图是否存在一个大小为k的完全子图
  - 完全子图：每两个不同的点之间都恰存在一条连线

分配问题（1）

- 将n个任务分配给n个人执行，每个人只执行一个任务，每个任务只能给一个人执行；将第j个任务分配给第i个人执行的成本是C[i,j]；如何找到总成本最小的分配方案？
- 二次分配问题
  - 成本计算的条件更加复杂：将j任务分配给第i人的同时将l任务分配给第k人的成本为C[I,j,k,l]
- 例. 有如下人员和任务对应的成本计算表格，求成本最小的分配方案

	任务1	任务2	任务3	任务4
人员1	9	2	7	8
人员2	6	4	3	7
人员3	5	8	1	8
人员4	7	6	9	4

- 穷举查找：列出所有可能的分配方案

分配问题（2）

- 确定人员顺序为1，2，3，4；则任务序列为所有任务的一种排列
- 计算所有排列对应的成本

{1,2,3,4}——9+4+1+4=18

{1,2,4,3}——9+4+8+9=30

{1,3,2,4}——9+3+8+4=24

{1,3,4,2}——9+3+8+6=26

{1,4,2,3}——9+7+8+9=33

{1,4,3,2}——9+7+1+6=23

……

- 排列的方式有n!个
- 一个更加高效的算法：匈牙利算法（两位匈牙利数学家的贡献）

分配问题（3）

- 匈牙利算法
  - 1. 构造成本矩阵
  - 2. 每行、每列减去最小值
  - 3. 以每一个独立的0画十字，已经在线上的0不独立
  - 4. 如果十字可以覆盖所有的行和列，则已经存在最优解，即每行、列找到一个独立的0，对应的位置为最优解（使用最少的直线覆盖所有0，如果直线数目<n则没有达到最优）；否则则进行下一步
  - 5. 用最少的直线覆盖所有0；没有被直线覆盖的区域每行减去区域内最小值，再执行覆盖区域每列加上该最小值，回到第3步

- 之前的例子：

$$\begin{bmatrix} 9 & 2 & 7 & 8 \\ 6 & 4 & 3 & 7 \\ 5 & 8 & 1 & 8 \\ 7 & 6 & 9 & 4 \end{bmatrix} \text{每行减去行内最小值：} \begin{bmatrix} 7 & 0 & 5 & 6 \\ 3 & 1 & 0 & 4 \\ 4 & 7 & 0 & 7 \\ 3 & 2 & 5 & 0 \end{bmatrix} \text{每列减去列上最小值} \begin{bmatrix} 4 & 0 & 5 & 6 \\ 0 & 1 & 0 & 4 \\ 1 & 7 & 0 & 7 \\ 0 & 2 & 5 & 0 \end{bmatrix}$$

$$\text{用直线覆盖：} \begin{bmatrix} 4 & 0 & 5 & 6 \\ 0 & 1 & 0 & 4 \\ 1 & 7 & 0 & 7 \\ 0 & 2 & 5 & 0 \end{bmatrix} \text{需要4条，等于行数，说明可以确定最优解} \begin{bmatrix} 4 & 0 & 5 & 6 \\ 0 & 1 & 0 & 4 \\ 1 & 7 & 0 & 7 \\ 0 & 2 & 5 & 0 \end{bmatrix}$$

分配问题（4）

- 例题：将4个推土机分配到四个不同工地

推土机 \ 工地	A	B	C	D
1	90	75	75	80
2	35	85	55	65
3	125	95	90	105
4	45	110	95	115

$$\begin{bmatrix} 90 & 75 & 75 & 80 \\ 35 & 85 & 55 & 65 \\ 125 & 95 & 90 & 105 \\ 45 & 110 & 95 & 115 \end{bmatrix} \rightarrow \begin{bmatrix} 15 & 0 & 0 & 5 \\ 0 & 50 & 20 & 30 \\ 35 & 5 & 0 & 15 \\ 0 & 65 & 50 & 70 \end{bmatrix} \rightarrow \begin{bmatrix} 15 & 0 & 0 & 0 \\ 0 & 50 & 20 & 25 \\ 35 & 5 & 0 & 10 \\ 0 & 65 & 50 & 65 \end{bmatrix} \rightarrow \begin{bmatrix} 15 & 0 & 0 & 0 \\ -20 & 30 & 0 & 5 \\ 35 & 5 & 0 & 10 \\ -20 & 45 & 30 & 45 \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} 35 & 0 & 0 & 0 \\ 0 & 30 & 0 & 5 \\ 55 & 5 & 0 & 10 \\ 0 & 45 & 30 & 45 \end{bmatrix} \rightarrow \begin{bmatrix} 35 & 0 & 0 & 0 \\ 0 & 30 & 0 & 5 \\ 55 & 5 & 0 & 10 \\ 0 & 45 & 30 & 45 \end{bmatrix} \rightarrow \begin{bmatrix} 35 & 0 & 0 & 0 \\ -5 & 25 & -5 & 0 \\ 50 & 0 & -5 & 5 \\ -5 & 40 & 25 & 40 \end{bmatrix} \rightarrow \begin{bmatrix} 40 & 0 & 5 & 0 \\ 0 & 25 & 0 & 0 \\ 55 & 0 & 0 & 5 \\ 0 & 40 & 30 & 40 \end{bmatrix}$$

例题——搜索解空间

- 假金币问题
  - 有N个金币，其中一个是假的，重量小于真金币，其它真金币的重量都相同；只有一台天平可以使用
  - 问题一：设计一个算法找到假金币
  - 问题二：给定一些测试的结果，判断假金币的编号；

左个数	右个数	大小	左编号	右编号
2	1	>	2，3	4
1	1	=	2	5
1	1	=	1	4

  - 根据规则推理？
  - 假设编号k的为假的，判断是否矛盾

  - 类似的题目：有ABCDEF六个犯罪嫌疑人，有如下结论，判断哪些是真的罪犯：

    1) A、B 至少有一人作案；
    2) A、E、F 这3 人中至少有两人参与作案；
    3) A、D 不可能是同案犯；
    4) B、C 或同时作案，或与本案无关；
    5) C、D 中有且仅有1 人作案；
    6) 如果D 没有参与作案，则E 也不可能参与作案。

蛮力之直接计算

- 计算N！有多少位？

- 斯特林公式（Stirling's approximation）是一条用来取n的阶乘的近似值的数学公式。一般来说，当n很大的时候，n阶乘的计算量十分大，所以斯特林公式十分好用，而且，即使在n很小的时候，斯特林公式的取值已经十分准确。

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

或更精确的

$$\lim_{n\to+\infty}\frac{n!}{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n} = 1$$

或

$$\lim_{n\to\infty}\frac{e^n\, n!}{n^n\sqrt{n}} = \sqrt{2\pi}$$

- 冲撞的机器人

输 入

第一行是 K，测试序列的组数。

每个测试序列开始一行是两个整数 A,B，1 <= A, B <= 100，给定了仓库的大小。A 是东西方向，而 B 是南北方向。第二行是两个整数 M,N,1 <= N, M <= 100，分别表示机器人的数目和指令的数目。随后 N 行包含 2 个整数，1 <= Xi <= A, 1 <= Yi <= B 和一个字母(N，S，E 或 W)，从 1 到 N 给定每个机器人的初始位置和方向，没有两个机器人的初始位置是一样的。

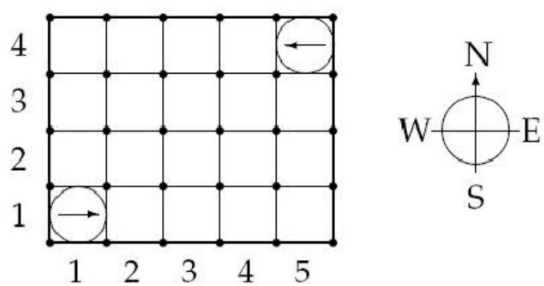


图 2-4-1 事例仓库中机器人的初始位置

最后 M 行，顺序给定所有指令。

指令具有如下格式：

< 机器人编号>< 操作><次数>

3 种操作：

L： 左转 90 度；

R： 右转 90 度；

F:： 向前一个单位。

1<=<次数> <= 100 是指令被机器人执行的次数。

输入样例

4
5 4
2 2
1 1 E
5 4 W
1 F 7
2 F 7
5 4
2 4
1 1 E
5 4 W
1 F 3
2 F 1
1 L 1
1 F 3
5 4
2 2
1 1 E
5 4 W
1 L 96
1 F 2
5 4
2 3
1 1 E
5 4 W
1 F 4
1 L 1
1 F 20

总结

- 蛮力法
  - 最简单直接的问题求解思路
  - 选择排序、冒泡排序
  - 顺序查找、蛮力字符串匹配
  - 最近对、凸包问题——根据问题定义
  - 穷举查找——找到需要穷举的项的概念
    - 旅行商问题
    - 背包问题
    - 分配问题
  - 直接计算
  - 模拟和仿真

- 下次课：分治法的其它问题

谢 谢！