

第五章 搜索与回溯算法

【上机练习】

1、全排列问题(form.cpp)

【问题描述】

输出自然数 1 到 n 所有不重复的排列，即 n 的全排列，要求所产生的任一数字序列中不允许出现重复的数字。

【输入格式】

$n(1 \leq n \leq 9)$

【输出格式】

由 $1 \sim n$ 组成的所有不重复的数字序列，每行一个序列。

【输入样例】

3

【输出样例】

```
1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1
```

2、组合的输出(compages.cpp)

【问题描述】

排列与组合是常用的数学方法，其中组合就是从 n 个元素中抽出 r 个元素(不分顺序且 $r \leq n$)，我们可以简单地将 n 个元素理解为自然数 $1, 2, \dots, n$ ，从中任取 r 个数。

现要求你用递归的方法输出所有组合。

例如 $n=5, r=3$ ，所有组合为：

1 2 3 1 2 4 1 2 5 1 3 4 1 3 5 1 4 5 2 3 4 2 3 5 2 4 5 3 4 5

【输入】

一行两个自然数 $n, r(1 < n < 21, 1 \leq r \leq n)$ 。

【输出】

所有的组合，每一个组合占一行且其中的元素按由小到大的顺序排列，每个元素占三个字符的位置，所有的组合也按字典顺序。

【样例】

compages.in	compages.out
5 3	1 2 3
	1 2 4
	1 2 5
	1 3 4
	1 3 5
	1 4 5
	2 3 4
	2 3 5
	2 4 5
	3 4 5

3、N 皇后问题(queen.cpp)

【问题描述】

在 $N \times N$ 的棋盘上放置 N 个皇后 ($n \leq 10$) 而彼此不受攻击 (即在棋盘的任一行，任一列和任一对角线上不能放置 2 个皇后)，编程求解所有的摆放方法。

	1	2	3	4	5	6	7	8
1						♛		
2			♛					
3					♛			
4							♛	
5	♛							
6				♛				
7		♛						
8								♛

	1	2	3	4	5	6	7	8
1								
2	♛							
3					♛			
4		♛						
5								♛
6			♛					
7							♛	
8				♛				

八皇后的两组解

【输入格式】

输入: n

【输出格式】

每行输出一种方案, 每种方案顺序输出皇后所在的列号, 各个数之间有空格隔开。若无方案, 则输出 no solute!

【输入样例】

4

【输出样例】

2 4 1 3
3 1 4 2

4、有重复元素的排列问题 (perm)

【问题描述】

设 $R=\{r_1, r_2, \dots, r_n\}$ 是要进行排列的n个元素。其中元素 r_1, r_2, \dots, r_n 可能相同。试设计一个算法, 列出R的所有不同排列。

【编程任务】

给定n 以及待排列的n 个元素。计算出这n 个元素的所有不同排列。

【输入格式】

由perm.in输入数据。文件的第1 行是元素个数n, $1 \leq n \leq 500$ 。接下来的1 行是待排列的n个元素。

【输出格式】

计算出的n个元素的所有不同排列输出到文件perm.out中。文件最后1行中的数是排列总数。

【输入样例】

4

aacc

【输出样例】多解

aacc

acac

acca

caac

caca

ccaa

6

5、子集和问题 (subsum)

【问题描述】

子集和问题的一个实例为 $\langle S, t \rangle$ 。其中, $S=\{x_1, x_2, \dots, x_n\}$ 是一个正整数的集合, c是一个正整数。子集和问题判定是否存在S的一个子集 S_1 , 使得子集 S_1 和等于c。

【编程任务】

对于给定的正整数的集合 $S=\{x_1, x_2, \dots, x_n\}$ 和正整数c, 编程计算S 的一个子集 S_1 , 使得子集 S_1 和等于c。

【输入格式】

由文件subsum.in提供输入数据。文件第1行有2个正整数n和c，n表示S的个数，c是子集和的目标值。接下来的1行中，有n个正整数，表示集合S中的元素。

【输出格式】

程序运行结束时，将子集和问题的解输出到文件subsum.out中。当问题无解时，输出“No solution!”。

【输入样例】

```
5 10
2 2 6 5 4
```

【输出样例】

```
2 2 6
```

6、工作分配问题(job)

【问题描述】

设有n件工作分配给n个人。将工作i分配给第j个人所需的费用为 c_{ij} 。试设计一个算法，为每一个人都分配一件不同的工作，并使总费用达到最小。

【编程任务】

设计一个算法，对于给定的工作费用，计算最佳工作分配方案，使总费用达到最小。

【输入格式】

由文件job.in给出输入数据。第一行有1个正整数n ($1 \leq n \leq 20$)。接下来的n行，每行n个数，第i行表示第i个人各项工作费用。

【输出格式】

将计算出的最小总费用输出到文件job.out。

【输入样例】

```
3
4 2 5
2 3 6
3 4 5
```

【输出样例】

```
9
```

7、装载问题(load)

【问题描述】

有一批共n个集装箱要装上艘载重量为c的轮船，其中集装箱i的重量为 w_i 。找出一种最优装载方案，将轮船尽可能装满，即在装载体积不受限制的情况下，将尽可能重的集装箱装上轮船。

【输入格式】

由文件load.in给出输入数据。第一行有2个正整数n和c。n是集装箱数，c是轮船的载重量。接下来的1行中有n个正整数，表示集装箱的重量。

【输出格式】

将计算出的最大装载重量输出到文件load.out。

【输入样例】

```
5 10
7 2 6 5 4
```

【输出样例】

```
10
```

8、字符序列(characts)

【问题描述】

从三个元素的集合[A, B, C]中选取元素生成一个 N 个字符组成的序列，使得没有两个相邻字的子序列（子序列长度=2）相同。例：N = 5 时 ABCBA 是合格的，而序列 ABCBC 与 ABABC 是不合格的，因为其中子序列 BC，AB 是相同的。

对于由键盘输入的 N($1 \leq N \leq 12$)，求出满足条件的 N 个字符的所有序列和其总数。

【输入样例】

```
4
```

【输出样例】

9、试卷批分(grade)

【问题描述】

某学校进行了一次英语考试，共有 10 道是非题，每题为 10 分，解答用 1 表示“是”，用 0 表示“非”的方式。但老师批完卷后，发现漏批了一张试卷，而且标准答案也丢失了，手头只剩下了 3 张标有分数的试卷。

试卷一：	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	
	0	0	1	0	1	0	0	1	0	0	得分：70
试卷二：	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	
	0	1	1	1	0	1	0	1	1	1	得分：50
试卷三：	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	
	0	1	1	1	0	0	0	1	0	1	得分：30
待批试卷：	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	
	0	0	1	1	1	0	0	1	1	1	得分：？

【问题求解】：

请编一程序依据这三张试卷，算出漏批的那张试卷的分数。

10、迷宫问题(migong)

【问题描述】

设有一个 $N*N$ ($2 \leq N < 10$) 方格的迷宫，入口和出口分别在左上角和右上角。迷宫格子中分别放 0 和 1，0 表示可通，1 表示不能，入口和出口处肯定是 0。迷宫走的规则如下所示：即从某点开始，有八个方向可走，前进方格中数字为 0 时表示可通过，为 1 时表示不可通过，要另找路径。找出所有从入口（左上角）到出口（右上角）的路径(不能重复)，输出路径总数，如果无法到达，则输出 0。

【输入样例】

```
3
000
011
100
```

【输出样例】

```
2          //路径总数
```

11、部落卫队

【问题描述】

原始部落 byteland 中的居民们为了争夺有限的资源，经常发生冲突。几乎每个居民都有他的仇敌。部落酋长为了组织一支保卫部落的队伍，希望从部落的居民中选出最多的居民入伍，并保证队伍中任何 2 个人都不是仇敌。

【编程任务】

给定 byteland 部落中居民间的仇敌关系，编程计算组成部落卫队的最佳方案。

【输入格式】

第 1 行有 2 个正整数 n 和 m ，表示 byteland 部落中有 n 个居民，居民间有 m 个仇敌关系。居民编号为 1, 2, ..., n 。接下来的 m 行中，每行有 2 个正整数 u 和 v ，表示居民 u 与居民 v 是仇敌。

【输出格式】

第 1 行是部落卫队的总人数；第 2 行是卫队组成 x_i ， $1 \leq i \leq n$ ， $x_i = 0$ 表示居民 i 不在卫队中， $x_i = 1$ 表示居民 i 在卫队中。

【输入样例】

```
7 10
1 2
```

```

1  4
2  4
2  3
2  5
2  6
3  5
3  6
4  5
5  6
【输出样例】
3
1 0 1 0 0 0 1

```

12、最佳调度问题

【问题描述】

假设有 n 个任务由 k 个可并行工作的机器完成。完成任务 i 需要的时间为 t_i 。试设计一个算法找出完成这 n 个任务的最佳调度，使得完成全部任务的时间最早。

【编程任务】

对任意给定的整数 n 和 k ，以及完成任务 i 需要的时间为 t_i ， $i=1\sim n$ 。编程计算完成这 n 个任务的最佳调度。

【输入格式】

由文件machine.in给出输入数据。第一行有2 个正整数 n 和 k 。第2 行的 n 个正整数是完成 n 个任务需要的时间。

【输出格式】

将计算出的完成全部任务的最早时间输出到文件machine.out。

【输入样例】

```

7 3
2 14 4 16 6 5 3

```

【输出样例】

```

17

```

13、图的 m 着色问题

【问题描述】

给定无向连通图 G 和 m 种不同的颜色。用这些颜色为图 G 的各顶点着色，每个顶点着一种颜色。如果有一种着色法使 G 中每条边的2个顶点着不同颜色，则称这个图是 m 可着色的。图的 m 着色问题是对于给定图 G 和 m 种颜色，找出所有不同的着色法。

【编程任务】

对于给定的无向连通图 G 和 m 种不同的颜色，编程计算图的所有不同的着色法。

【输入格式】

第1行有3个正整数 n ， k 和 m ，表示给定的图 G 有 n 个顶点和 k 条边， m 种颜色。顶点编号为1，2， \dots ， n 。接下来的 k 行中，每行有2个正整数 u ， v ，表示图 G 的一条边 (u, v) 。

【输出格式】

将计算出的不同的着色方案数输出。

【输入样例】

```

5 8 4
1 2
1 3
1 4
2 3
2 4
2 5
3 4
4 5

```

【输出样例】

```

48

```