第8章 课后习题

Q1：阿克曼(Ackermann)函数	Q6：根据积分值反推积分上界
Q2：计算连分数$f(x,n)$	Q7：求定积分的最大值
Q3：生成范德蒙矩阵	Q8：fminsearch优化测试函数
Q4：Fisher-Yates shuffle洗牌算法	Q9：计算给定坐标点之间的距离矩阵
Q5：数值积分的练习题	Q10：生成扫雷游戏的地图

Q1：阿克曼(Ackermann)函数

阿克曼(Ackermann)函数是一个经典的函数，通常用于研究计算复杂性。它是一个二元函数，通常定义为：

$$y(m,n)=\begin{cases} n+1 & (m=0\text{ 时}) \\ y(m-1,1) & (m>0,n=0\text{ 时}) \\ y(m-1,\ y(m,n-1)) & (m,n>0\text{ 时}) \end{cases}$$

其中，m和n的定义域是非负整数(请注意，阿克曼函数的值增长非常快，对于相对较小的m和n值，它可能会产生非常大的结果。因此，大家在测试时，可以取m≤3、n≤10)

请编写一个MATLAB函数ackermann来计算阿克曼函数的值（右侧有供参考的调用结果）。

```
ackermann(3, 6)

ans = 509

ackermann(2, 3)

ans = 9
```

Q2：计算连分数$f(x,n)$

已知：$$f(x,n)=\cfrac{x}{n+\cfrac{x}{(n-1)+\cfrac{x}{(n-2)+\ddots+\cfrac{x}{1+x}}}}$$

式中n是一个正整数，x是一个正数。
编写一个MATLAB函数fun计算$f(x,n)$。

例如：$$f(1.5,4)=\cfrac{1.5}{4+\cfrac{1.5}{3+\cfrac{1.5}{2+\cfrac{1.5}{1+1.5}}}}\approx 0.3394$$

```
fun(1.5, 4)

ans = 0.3394

fun(3, 10)

ans = 0.2907

fun(3, 1)

ans = 0.7500
```

进阶：你能使用递归和非递归两种方法计算吗？

Q3：生成范德蒙德矩阵

本题借助文心一言工具辅助生成

定义一个名为 vdm 的 MATLAB 函数，用于生成一个 n 阶范德蒙德矩阵 A_n。该函数设计为接受以下两种输入模式：

注：有的教材也译为范德蒙矩阵

1. **单向量输入**：函数接受一个数值向量 x 作为输入参数，该向量可以是水平向量 $x = [x_1, x_2, \ldots, x_n]$ 或垂直向量 $x = [x_1; x_2; \ldots; x_n]$。
2. **多标量输入**：函数接受 n 个数值标量 $x_1, x_2, \ldots, x_n$ 作为独立的输入参数。

范德蒙德矩阵 A_n 的结构定义如下：

$$A_n = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix}$$

函数 vdm 应满足以下要求：

假设用户输入的数据中至少包含两个常数。

- 支持至少 n=2 的情况，即至少生成一个 2x2 的范德蒙德矩阵。
- 忽略错误输入参数的检查，假设所有输入都是有效的。

```
A1 = vdm(1,2,3)

A1 = 3×3
     1     1     1
     1     2     4
     1     3     9
```

```
A2 = vdm([2 -3 3 1])

A2 = 4×4
     1     2     4     8
     1    -3     9   -27
     1     3     9    27
     1     1     1     1
```

```
A3 = vdm([-1;-2;3;5])

A3 = 4×4
     1    -1     1    -1
     1    -2     4    -8
     1     3     9    27
     1     5    25   125
```

Q4：Fisher-Yates shuffle洗牌算法

Fisher-Yates shuffle洗牌算法是一种生成随机排列的算法。该算法可以保证每个排列出现的概率是相等的，因此它是一种公平的洗牌方法。

请大家在网上搜索该算法的步骤，编写一个名为my_shuffle的函数，它能对1,2,...,n的序列进行随机的打乱。

（拓展：在第三章的课后习题中，我们详细介绍过randperm函数的用法，例如randperm(n)就是对1,2,...,n的序列进行随机的打乱，因此my_shuffle(n)和randperm(n) 的功能一致）

```
my_shuffle(5)
```

```
ans = 1×5
     1     4     3     2     5
```

```
my_shuffle(3)
```

```
ans = 1×3
     3     1     2
```

注意：由于算法的实现过程中用到了随机数，因此调用函数后每次得到的结果可能都不同。

Q5：数值积分的练习题

提示：本题考察“计算参数化函数的定积分”的内容

（1）编写一个脚本或函数，能根据给定的 a 值计算 $\int_1^{+\infty} \frac{1}{x(x+a)}\, dx$ 的数值积分，式中a是一个正整数。（参考的解析解：$\frac{\ln(a+1)}{a}$）

（2）不使用循环语句，计算当a=1,2,...,10时，上一问积分的计算结果，并将结果保存到包含10个元素的向量d中。

（3）编写一个脚本或函数，能根据给定的 a 和 b 的值来计算二重积分 $\int_0^b dy \int_{2y}^a e^{-x+y} dx$ 的数值积分，式中a和b均是正整数。（参考的解析解：$1-e^{-a}(e^b-1)-e^{-b}$ ）

Q6：根据积分值反推积分上界

设 $\int_0^a x e^{2x} dx = \frac{1}{4}$，则 $a =$ ________。

本题来自2014年考研数学三真题，答案是0.5，你能使用MATLAB得到这个结果吗？

提示：可以将本题转换为求零点的问题。

Q7：求定积分的最大值

已知$S(t)=\int_t^{2t} xe^{-2x}dx$，且$t>0$，那么$t$取何值时，$S(t)$最大?

解析解：当$t=ln\,2$时取到的最大值$\frac{ln\,2}{16}+\frac{3}{64}$

大家可以验证自己求得的数值解和解析解是否一致

提示：将S(t)视为关于t的一元函数，调用MATLAB的内置函数求它的最大值。

Q8：fminsearch优化测试函数

以n等于10为例（即该函数有10个自变量：x_1至x_{10}），求出第二个和第三个函数的最小值。

函数名	函数表达式
Sphere	$f(x)=\sum_{i=1}^{n}x_i^2$
Rastrigin	$f(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)$
Griewank	$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$
Rosenbrock	$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$

Q9：计算给定坐标点之间的距离矩阵

距离的计算公式可参考第三章的内容，函数的调用结果请参考下一页PPT

编写一个函数 mydist，用于计算给定坐标点之间的距离。

函数 mydist 可以接收以下三个输入参数：

（1）坐标矩阵 X：一个 n 行 k 列的数值矩阵，表示 n 个位置的坐标数据，其中 k=2 表示二维坐标，k=3 表示三维坐标。

（2）距离计算方法 Distance（可选）：一个字符向量，用于指定计算距离的方式。可选值如下：

- 'euclidean'（默认）：表示使用欧几里得距离。
- 'cityblock'：表示使用曼哈顿距离（又称城市街区距离）。
- 'minkowski'：表示使用闵可夫斯基距离。

（3）闵可夫斯基距离参数 p（可选）：当选择闵可夫斯基距离时，此参数指定公式中的 p 值。如果未指定该参数，则默认 p 等于 2。

函数的返回值 D 是一个 n 行 n 列的矩阵，表示各位置之间的距离。矩阵 D 是对称的，即 D(i,j) 等于 D(j,i)，且 D(i,i) 为 0，表示任一位置到自身的距离。

Q9：计算给定坐标点之间的距离矩阵

```
X = [1 2 3;
     2 5 8;
     3 6 9;
     4 11 0];
mydist(X)
```

```
ans = 4×4
         0    5.9161    7.4833    9.9499
    5.9161         0    1.7321   10.1980
    7.4833    1.7321         0   10.3441
    9.9499   10.1980   10.3441         0
```

```
mydist(X,'euclidean')
```

```
ans = 4×4
         0    5.9161    7.4833    9.9499
    5.9161         0    1.7321   10.1980
    7.4833    1.7321         0   10.3441
    9.9499   10.1980   10.3441         0
```

```
mydist(X,'cityblock')
```

```
ans = 4×4
     0     9    12    15
     9     0     3    16
    12     3     0    15
    15    16    15     0
```

```
mydist(X,'minkowski')
```

```
ans = 4×4
         0    5.9161    7.4833    9.9499
    5.9161         0    1.7321   10.1980
    7.4833    1.7321         0   10.3441
    9.9499   10.1980   10.3441         0
```

```
mydist(X,'minkowski',3)
```

```
ans = 4×4
         0    5.3485    6.6039    9.2170
    5.3485         0    1.4422    9.0287
    6.6039    1.4422         0    9.4912
    9.2170    9.0287    9.4912         0
```

```
mydist(X,'cityblock',3)
```

错误使用 q9>mydist
只有minkowski距离才能指定三个输入参数

Q10：生成扫雷游戏的地图

函数的调用结果可参考下一页

编写一个MATLAB函数 minesweeper，该函数用于随机生成扫雷游戏的地图。

输入参数：

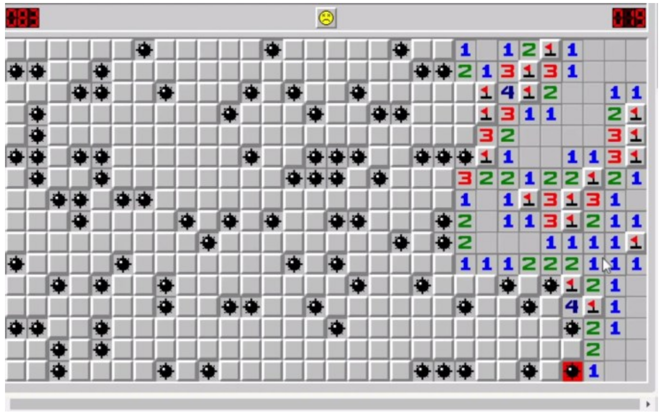
m：地图的行数（正整数）

n：地图的列数（正整数）

k：地图中雷的个数（正整数且$k \le m \times n$）

输出参数：

d：一个 m 行 n 列的矩阵，代表扫雷游戏的地图。



地图规则：

地图中的每个单元格可以包含两种值：Inf 或一个非负整数。

- Inf 表示该单元格下有雷。
- 非负整数表示该单元格四周8个邻居（如果有的话）中雷的总数。如果单元格位于边缘，邻居的数量会少于8个，例如左上角只有3个邻居。

Q10：生成扫雷游戏的地图

```
d = minesweeper(5,6,8)
```

```
d = 5×6

     0     1   Inf   Inf     3     2
     0     1     2     3   Inf   Inf
     0     0     0     1     2     2
     1     2     3     3     2     1
     1   Inf   Inf   Inf   Inf     1
```

```
d = minesweeper(10,5,15)
```

```
d = 10×5

   Inf     4   Inf     2     0
   Inf     5   Inf     4     1
     1     3   Inf     3   Inf
     1     2     1     2     1
   Inf     3     1     1     1
   Inf   Inf     2     3   Inf
     2     2     2   Inf   Inf
     0     0     1     2     2
     1     1     1     2     2
   Inf     1     1   Inf   Inf
```