

Assignment 01

1. Flowchart

在题目中，根据流程图的目的，其通过两两比较大小对数字 a、b、c 进行排序。对此，提供了三种方案：（1）完全依照流程图进行两两比较，并最终返回结果；（2）通过 Python 的链式比较写法，同时判断三个数字的大小；（3）使用列表的 sort 方法，返回大小递减的数组。

2. Continuous celing function

题目要求编写一个调用自身（递归）的函数，因此设定好递归返回条件（ $F(1) = 1$ ）防止无限计算下去即可。

3. Dice rolling

3.1

对于计算 10 个骰子获得总点数 x 的途径总数。可以考虑直接计算所有组合，不过计算量会比较大。因此这里通过递归的方式将目标分解，设计函数：将目标骰子总面值 x 和剩余骰子数 n 作为参数，计算单个骰子取 1~6 时剩余 n-1 个骰子总面值为 x-1 ~ x-6 的途径数（这个过程可以递归实现），最终在仅有 1 个骰子时终止递归，并不断累加途径数量。

3.2

在 10 个骰子，设置 x 取 10 到 60 时，可以看到结果非常接近正态分布，根据中心极限定理，随着骰子数目增加，投出的平均值将集中的 3.5 附近，也即在投出该值的可能性最大（途径数最多），对应的总点数即为 $3.5 \times n$ ，在 n 取 10 时，x 取 35。程序计算的结果也支持该结论。

4. Dynamic programming

4.1

需要获得长度为 N 大小介于 0 到 10 的整数随机数组。因此使用 random 模块的 random 函数获取从 0 到 11 的浮点数，然后向下取整，重复 N 次获得最终的数组。使用的函数决定当 N 较大时，结果在 0 到 10 之间均匀分布。

4.2

需要书写函数计算随机数组的所有子集平均值的和。

一种方式比较简单，使用 itertools 库的 combinations 方法，对数组的所有子集进行迭代，需要计算 $\sum_{i=1}^N C_N^i = \sum_{i=1}^N \frac{N!}{i!(N-i)!}$ 次，当 N 非常大时则计算量非常庞大。

另一种方式先从数学上对计算过程进行简化。因为计算的是数组的所有子集，因此数组中的每个元素应当是等价的，也即其对最终结果的贡献权重应当是相等的。此时我们计算出该权重再乘以数组和即可得到最终结果。其公式书写如下：

$$\text{Sum} = \sum_{i=1}^N \left(\frac{1}{i} \times \frac{N!}{i!(N-i)!} \times \frac{i}{N} \right) \sum_{j=1}^N x_j = \sum_{i=1}^N \left[\frac{N!}{i!(N-i)!} \right] \sum_{j=1}^N \frac{x_j}{N} \quad (1)$$

使用程序实现，阶乘通过 math 库的 factorial 函数计算，然后使用 sum 对列表进行求和即可。

对于 4.1 定义随机数组的方式，可以得到理论平均值在 5 附近，因此可以得到理论结果为

$$\text{Sum} = 5 \times \sum_{i=1}^N \frac{N!}{i!(N-i)!}。$$

4.3

计算了 N 从 1 到 100 的结果，并将结果绘制出来（这里对 y 轴取了对数），可以看到随着 N 增大，所有子集平均值的和也指数增加。根据集合论的基本定理，n 个元素的集合子集总数为 2^n （该定理通过谷歌搜索获得）。由于这里不考虑非空子集，且在 n 较大时平均值趋向于 5，所以得到的理想结果应该在

$$\text{Sum} = 5 \times \sum_{i=1}^N \frac{N!}{i!(N-i)!} = 5 \times (2^N - 1) \text{ 附近。}$$

5. Path counting

5.1

需要创建一个二维数组，并且需要在保证左上和右下设置为 1 的同时，在其他位置随机填充 0 和 1。虽然可以直接从 0 和 1 中抽取，不过考虑到题目是路径计算，障碍物的密度也会影响到计算结果，因此采用先抽取 0 到 1 的浮点数，然后设置阈值，高于高阈值则设为 1，否则设为 0。最后在手动设置左上和右下位置为 1。

5.2

由于题目要求只能朝右下前进，因此对于任意路径，其长度固定为 $M + N - 2$ ，也就是向右 $M - 1$ 次，向下 $N - 1$ 次。那么可以采取排列组合的方式，在长度 $M + N - 2$ 的位移次数中，抽取 $M - 1$ 个位置向右， $N - 1$ 个位置向下，此时该路径上如果没有障碍物，则可行路径加一。

5.3

在 10 行 8 列，30 % 网格存在障碍物的情况下，1000 次运行平均可能路径数为 57.231