Notes:

• Present the solutions (answers, mathematical formulas, Python code, results, etc.) in a Jupyter Notebook file.

• Verify all results using simulation methods.

• Any calculations and reasoning should be justified with theoretical arguments whenever possible.

EXERCISE 1

COMPUTER STATISTICS AND APPLICATIONS

Question 1. (3 points)

An and Bình are playing a game consisting of multiple rounds. The first player to reach 10 points wins the game. Each round has only one winner. Assume the rounds are independent, and An wins each round with a probability of 0.6.

a) Calculate the probability that An wins the game if the winner of each round earns 2 points.

b) Suppose instead that if the winner has already won a round immediately before, they only earn 1 point.

c) Determine the expected number of rounds that An plays in cases (a) and (b).

Question 2. (2.5 points)

Let U be a random variable uniformly distributed over (0,1).

a) For constants a < b, determine the distribution of $a + (b - a)U$.

b) For constants $0 < a < b < 1$, determine the distribution of U given $a < U < b$.

c) Given that V is a random variable uniformly distributed over (0,1), determine the distribution of max {U, V}.

Question 3. (2.5 points)

Choose two random points A, B in $S = [0,1]^k$. Let D be the distance between A and B.

a) Compute E(D) when k = 1.

b) Compute E(D) when k = 2.

c) Analyze $r_k = E(D)/\sqrt{k}$ for k = 1, 2, ..., 10.

Consider an array a[1..n] containing half 0s and half 1s. The task is to find a position in a where the value is 1. Consider the following two algorithms:

Algorithm 1

```
i = 1
while a[i] != 1:
  i++
return i
```

Algorithm 2

```
while True:
  i = random(1, n)
  if a[i] == 1:
    return i
```

The function random(1, n) returns a random integer between 1 and n.

a) Compare the two algorithms.

b) Which algorithm is more efficient?