Question 1

1. A coin is tossed infinitely many times. For every $n = 2, 3, 4, \ldots$, let A_n be the event "a block of heads of length n starting from toss n " (that is heads at tosses $n, \ldots, 2n-1$, tails at tosses $n-1$ and $2n$).

a. Find the probability of A_n

Solution.

Let H_i (resp. T_i) denote "head (resp. tail) at toss i." Then

$$A_n \;=\; T_{n-1} \,\cap\, \Big(\bigcap_{i=n}^{2n-1} H_i\Big) \,\cap\, T_{2n}.$$

By independence of coin tosses,

$$\mathbb{P}(A_n) = \mathbb{P}(T_{n-1}) \cdot \prod_{i=n}^{2n-1} \mathbb{P}(H_i) \cdot \mathbb{P}(T_{2n}) = \left(\tfrac{1}{2}\right) \cdot \left(\tfrac{1}{2}\right)^n \cdot \left(\tfrac{1}{2}\right) = 2^{-(n+2)}.$$

□

b. Is A_n an increasing sequence?

Solution.

No. For an increasing sequence we would need $A_n \subseteq A_{n+1}$ for all n. But A_n requires T_{2n} while A_{n+1} requires H_{2n}, so $A_n \cap A_{n+1} = \varnothing$. Hence $A_n \not\subseteq A_{n+1}$.

□

c. Is A_n a decreasing sequence?

Solution.

No. A decreasing sequence would satisfy $A_{n+1} \subseteq A_n$ for all n, which again fails since $A_n \cap A_{n+1} = \varnothing$ and both events are nonempty.

□

d. Is A_n a convergent sequence?

Solution.

Recall that $\liminf_n A_n = \bigcup_m \bigcap_{n \geq m} A_n$ ("A_n ultimately") and $\limsup_n A_n = \bigcap_m \bigcup_{n \geq m} A_n$ ("A_n infinitely often"). Because $A_n \cap A_{n+1} = \varnothing$, it is impossible that A_n occurs for all sufficiently large n, so $\liminf_n A_n = \varnothing$.

On the other hand, one can construct sequences where A_n occurs for infinitely many n (e.g. arrange disjoint blocks TH^nT spaced far apart), so $\limsup_n A_n \neq \varnothing$.

Thus $\liminf_n A_n \neq \limsup_n A_n$ and (A_n) does not converge.

□

e. What is the probability that A_n occurs infinitely often?

Solution.

We have

$$\sum_{n=2}^{\infty} \mathbb{P}(A_n) = \sum_{n=2}^{\infty} 2^{-(n+2)} = \tfrac{1}{8} < \infty.$$

By the First Borel–Cantelli lemma, $\mathbb{P}(A_n \text{ i.o.}) = 0$.

□

f. What is the probability that A_n occurs ultimately?

Solution.

Since $\mathbb{P}(A_n) = 2^{-(n+2)} \to 0$ and $\mathbb{P}(\liminf_n A_n) \leq \liminf_{n\to\infty} \mathbb{P}(A_n)$, we get

$$\mathbb{P}(A_n \text{ ultimately}) = \mathbb{P}(\liminf_n A_n) = 0.$$

(Equivalently, $\liminf_n A_n \subseteq \limsup_n A_n$ and part (e) already gave $\mathbb{P}(\limsup_n A_n) = 0$.)

□

Question 2

2. Simulate in Matlab or in Python 1000 tossing of a fair coin.

1. Count the number of heads and the number of tails.

Solution.

```python
import numpy as np

rng = np.random.default_rng(40313)  # fixed seed for reproducibility
n = 1000

# True = Head, False = Tail
tosses = rng.random(n) < 0.5

heads = int(tosses.sum())
tails = n - heads

print(f"Heads: {heads}, Tails: {tails}")
```

Heads: 477, Tails: 523

□

2. Find the frequency of heads as a function of n ($n = 1, \dots, 1000$) and represent the function.

Solution.

```python
import numpy as np
import matplotlib.pyplot as plt

rng = np.random.default_rng(2024)
n = 1000

# 1 = Head, 0 = Tail
tosses = (rng.random(n) < 0.5).astype(int)

cum_heads = np.cumsum(tosses)
n_vals = np.arange(1, n + 1)
freq = cum_heads / n_vals

plt.figure()
plt.plot(n_vals, freq, label="Observed head frequency")
plt.axhline(0.5, linestyle="--", linewidth=1, label="Theoretical p=0.5")
plt.xlabel("n (number of tosses)")
plt.ylabel("Frequency of heads up to n")
plt.title("Cumulative frequency of heads (fair coin)")
plt.legend()
plt.show()
```

□

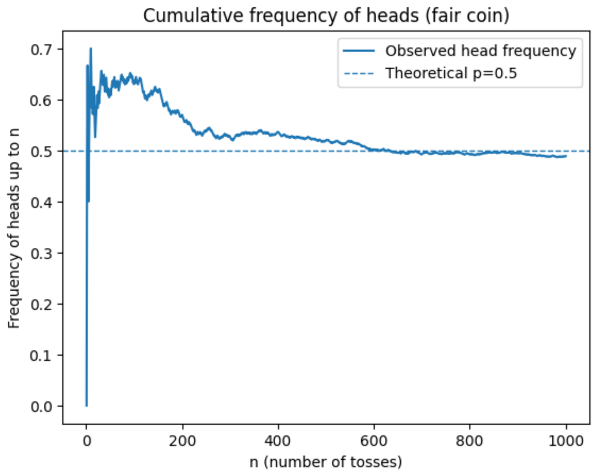

Figure 0-1: Cumulative Frequency of Heads (Fair Coin)

3. Count the number of blocks of heads of length 2 (that is sequences (THHT)).

Solution.

```python
import numpy as np

def count_THHT(tosses):
    """
    Count occurrences of the pattern Tail-Head-Head-Tail (THHT).
    Accepts either:
      - a string like "THHTHT...", or
      - an iterable of "H"/"T" characters.
    """
    s = tosses if isinstance(tosses, str) else "".join(tosses)
    return sum(1 for i in range(len(s) - 3) if s[i:i+4] == "THHT")

rng = np.random.default_rng(40313) # fixed seed for reproducibility
n = 1000

tosses = np.where(rng.random(n) < 0.5, "H", "T")  # array of "H"/"T"

print("THHT count:", count_THHT(tosses))
```

THHT count: 60

□

4. Repeat the above calculations for a coin with probability of heads=3/4.

Solution.

```python
import numpy as np
import matplotlib.pyplot as plt

# --- Setup ---
rng = np.random.default_rng(2024)
n = 1000
p = 0.75  # probability of Heads

# Simulate: 1 = Head, 0 = Tail
tosses = (rng.random(n) < p).astype(int)

# (a) Counts
heads = int(tosses.sum())
tails = n - heads
```

```
print(f"Heads: {heads}, Tails: {tails}")

# (b) Frequency of heads vs n
cum_heads = np.cumsum(tosses)
n_vals = np.arange(1, n + 1)
freq = cum_heads / n_vals

plt.figure()
plt.plot(n_vals, freq, label="Observed head frequency")
plt.axhline(p, linestyle="--", linewidth=1, label=f"Theoretical p={p}")
plt.xlabel("n (number of tosses)")
plt.ylabel("Frequency of heads up to n")
plt.title("Cumulative frequency of heads (biased coin, p=0.75)")
plt.legend()
plt.show()

# (c) Count occurrences of the pattern THHT
def count_THHT(tosses_ht_or_str):
    """
    Count occurrences of Tail-Head-Head-Tail (THHT).
    Accepts a string of 'H'/'T' or an iterable of 'H'/'T'.
    """
    s = tosses_ht_or_str if isinstance(tosses_ht_or_str, str) else "".join(tosses_ht_or_str)
    return sum(1 for i in range(len(s) - 3) if s[i:i+4] == "THHT")

# Convert to 'H'/'T' and count
ht = np.where(tosses == 1, "H", "T")
thht_count = count_THHT(ht)
print(f"Occurrences of 'THHT': {thht_count}")
```

Heads: 740, Tails: 260

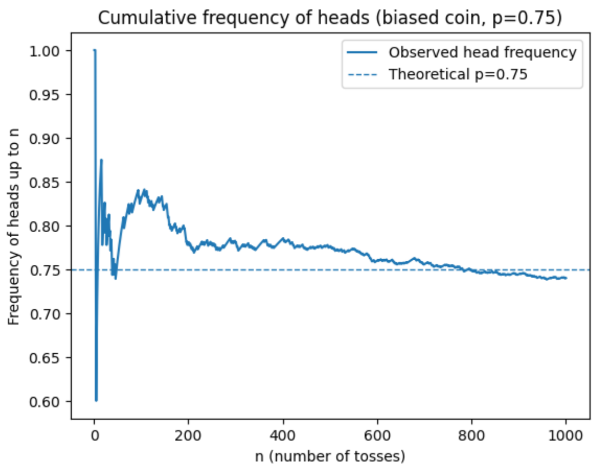

Figure 0-2: Cumulative Frequency of Heads (Biased Coin)

Occurrences of 'THHT': 46

□