Exercise 3

Due: 03/04/2019

Probability

We denote by "p-coin" a coin with p probability to get H (Heads) for $0 < p < 1$.

When we say that we flip a coin several times we assume the flips are independent of one another.

Question 1

1. Let there be a p-coin and a q-coin. We flip both coins. What is the probability that for the first we get H and for the second T?
2. We flip a p-coin until the first time in which we get H. What is the probability that the number of flips we perform is n? (Meaning at the n'th flip we get H for the first time).
3. We flip a p-coin n times. What is the probability we get T in the first k flips and H in the next $n-k$ flips?
4. Let $< \Omega, P >$ be a probability space. Prove from the definitions: for any two disjoint events $A, B \subseteq \Omega$ it holds that $P(A \cup B) = P(A) + P(B)$.
5. We flip a p-coin n times. What is the probability to get $n-k$ times H and k times T (where the order doesn't matter)?
   Guidance: use the previous question and the following definition.
   Definition: the binomial coefficient $\binom{n}{k}$ is the number of sub-sets of size k out of a set of size n.

Question 2

1. Prove that for any probability space $< \Omega, P >$ and an event $A \subseteq \Omega$ it holds that $\mathbb{E}[X_A] = P(A)$ where X_A is the indicator random variable of A (see the recitation summary for a definition).

2. Let $x_1, x_2, \ldots, x_n$ be a (finite) sequence of results of n flips of a p-coin. How many H do we get in expectation ($\mathbb{E}$)?
   Hint: define a sequence of indicator random variables and use the linearity of expectation.

3. Let $< \Omega, P >$ be a probability space, and let X and Y be two random variables for it.

   (a) Prove that if X and Y are independent, it holds that $\mathbb{E}[X \cdot Y] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$.
   Definition: Random variables X,Y are **independent** if for any $A, B \subseteq \mathbb{R}$ it holds that $P(\{X \in A\} \cap \{Y \in B\}) = P(\{X \in A\}) \cdot P(\{Y \in B\})$
   Hint: For the purpose of this subsquestion it suffices to use the fact that for independent random variables it holds that $P(X = x \wedge Y = y) = P(X = x) \cdot P(Y = y)$ for every $x, y \in \mathbb{R}$.
   Hint: use the definition $\mathbb{E}[X] = \sum_{x \in image(X)} x \cdot P(X = x)$ and note that $image(X \cdot Y) = \{xy | x \in image(X), y \in image(Y)\}$

   (b) Show a counterexample for the equation in subsection (a), for X and Y that are not independent.

Question 3

Prove that for the "Krav-Rav game" described at recitation 3 (and appearing in the summary), the event A_{i_0} that a specific game $1 \leq i_0 \leq n$ was drawn in the lottery and the event B_{x_0} that a specific value x_0 was the result of the dice after being thrown are independent.

Asymptotic Notations

Question 4

1. Prove the following using the definition of the asymptotic notations:

   (a) $\log(n) = o(n^\epsilon)$ for any $\epsilon > 0$ (hint: use L'Hopital rule)

   (b) $k_1^n = o(k_2^n)$ for any $k_1, k_2 \in \mathbb{N}$ satisfying $k_1 < k_2$

2. Order the following functions by dividing them into columns of asymptotically equivalent functions (they are θ of each other) such that functions in a left column are o of functions in a column to their right:

- n
- 2^n
- n^4

- $n^{1/3}$
- $n^4 - n^{3.6}$
- $\log_4 \log_2 n$
- $\log_3 n$
- $\log_2 n$
- 2^{n+2}

Quick select

Question 5

denote the following algorithm

```
Quick_select(arr, left, right, k):
    if left = right
        return arr[left]
    q = Partition(arr, left, right)
    if k == q:
        return arr[k]
    elif k < q:
        return Quick_select(arr, left, q-1, k)
    else:
        return Quick_select(arr, q+1, right, k)

Partition(arr, left, right):
    x = arr[right]
    i = left-1
    for j = left to right-1
        if arr[j] ≤ x
            i++
            swap(arr[j], arr[i])
    swap(arr[i+1], arr[right])
    return i+1
```

1. Illustrate the run of this algorithm when called by Quick_select(arr, 1, 8, 4) (for k=4) where arr is the following array: [7,2,3,0,5,6,1,4]
2. What does this algorithm return? What does it do?
3. Prove its correctness.

4. Is its worst case running time equal to its average running time? (don't need to write an answer just think of it)

5. Write a recursion formula for the worst case running time of Quick_select. Explain shortly why this is the formula for the worst case running time.

6. Find a tight asymptotic bound for the recursion formula you found in the previous subquestion and prove it.

7. In this subquestion we will find the the average running time for this algorithm:
   Assume PARTITION always returns an index $q \in Q_2 \bigcup Q_3$in the middle 2 quarters of the relevant sub-array, as in the following figure

   Figure 1:

Q1	Q2	Q3	Q4

   where the depicted array is the sub-array for which PARTITION was called (i.e. this is arr[left,..,right]).
   Construct the appropriate recursion formula given this assumption and solve it (i.e. find the running-time's asymptotic behaviour).