

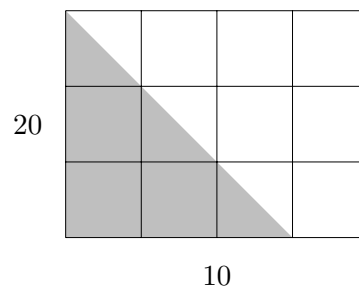
Mathematik für Informatiker
Kombinatorik, Stochastik und Statistik
Übungsblatt 2

Tom Paßberg , Iain Dorsch

Aufgabe 1

a)

Das Problem lässt sich mit der Formel aus Aufgabe 1 b) lösen. Hierzu setzen wir die Anzahl der Avenues auf 5 und die Anzahl der Streets auf 4. Die kürzesten Pfade repräsentieren die möglichen Anordnungen der Warteschlangen. Geht ein Pfad im ersten Schritt nach rechts, ist der erste Kunde ein Kunde mit einem 10€ Schein. Geht ein Pfad im ersten Schritt nach unten, ist der erste Kunde ein Kunde mit einem 20€ Schein. Ein Pfad ist gültig wenn er immer mindestens genauso oft nach rechts wie nach unten geht, er muss also über der Diagonale (siehe Abbildung) liegen.



Nach der Formel aus Aufgabe 1 b) ergibt sich also:

$$\binom{n+m}{n} - \binom{n+m}{n+1} = \binom{7}{4} - \binom{7}{5} = 14$$

b)

Aufgabe 2

Identität $a \rightarrow a$

$$\text{Drehung um } 72^\circ \quad a \rightarrow \begin{cases} a+1 & \text{if } x < 5 \\ 1 & \text{if } x = 5 \end{cases}$$

$$\text{Drehung um } 144^\circ \quad a \rightarrow \begin{cases} a+2 & \text{if } x < 4 \\ a-3 & \text{if } x \geq 4 \end{cases}$$

$$\text{Drehung um } 216^\circ \quad a \rightarrow \begin{cases} a+3 & \text{if } x < 3 \\ a-2 & \text{if } x \geq 3 \end{cases}$$

$$\text{Drehung um } 288^\circ \quad a \rightarrow \begin{cases} a+4 & \text{if } x < 2 \\ a-1 & \text{if } x \geq 2 \end{cases}$$

$$\text{Spiegelung 1 } a \rightarrow \begin{cases} 1 & \text{if } x = 1 \\ 5 & \text{if } x = 2 \\ 4 & \text{if } x = 3 \\ 3 & \text{if } x = 4 \\ 2 & \text{if } x = 5 \end{cases}$$

$$\text{Spiegelung 2 } a \rightarrow \begin{cases} 3 & \text{if } x = 1 \\ 2 & \text{if } x = 2 \\ 1 & \text{if } x = 3 \\ 5 & \text{if } x = 4 \\ 4 & \text{if } x = 5 \end{cases}$$

$$\text{Spiegelung 3 } a \rightarrow \begin{cases} 5 & \text{if } x = 1 \\ 4 & \text{if } x = 2 \\ 3 & \text{if } x = 3 \\ 2 & \text{if } x = 4 \\ 1 & \text{if } x = 5 \end{cases}$$

$$\text{Spiegelung 4 } a \rightarrow \begin{cases} 2 & \text{if } x = 1 \\ 1 & \text{if } x = 2 \\ 5 & \text{if } x = 3 \\ 4 & \text{if } x = 4 \\ 3 & \text{if } x = 5 \end{cases}$$

$$\text{Spiegelung 5 } a \rightarrow \begin{cases} 4 & \text{if } x = 1 \\ 3 & \text{if } x = 2 \\ 2 & \text{if } x = 3 \\ 1 & \text{if } x = 4 \\ 5 & \text{if } x = 5 \end{cases}$$

Die Spiegelungen 1, ..., 5 sind die Spiegelungen an den Geraden, die durch die Ecke und den Mittelpunkt des Fünfecks verlaufen.

Aufgabe 3

a)

Die Warscheinlichkeit, dass kein Brief beim Richtigen Empfänger ankommt ist 37,5%. Die Anzahl der Permutaionen ist $4! = 24$. Die Anzahl der Permutationen, bei denen kein Brief beim Richtigen Empfänger ankommt ist 9, das ergibt sich aus der Formel für die Anzahl der fixpunktfreien Permutationen.

1. $n = 1$:
 $a(n) = 0$

2. $n = 2$:
 $a(n) = 1$

3. $n = 3$:
 $a(n) = 2$

4. $n = 4$:
 $a(n) = 9$

Die Warscheinlichkeit ergibt sich aus

$$\frac{9}{4!} = 0.375$$

b)

$$a(n) = \begin{cases} 0 & \text{if } n = 1 \\ n \cdot a(n-1) + (-1)^n & \text{if } n > 1 \end{cases}$$

Aufgabe 4

a)

b)

c)

Code:

```
let sample_size = 100000;

let result = (6..).find(|&rounds| {
  let win_count = (0..sample_size).filter(|_|
    (0..rounds)
    .map(|_| rand::random::lt(usize::MAX) % 6)
    .unique()
    .count() == 6
  )
  .count();
  let win_rate = win_count as f64 / sample_size as f64;
  println!("Round: {}, Win rate: {}", rounds, win_rate);
  win_rate > 0.5
}).unwrap();

println!("Ab {} Runden hat der Spieler einen Vorteil.
Die Bank darf demnach maximal n = {} wahlen.",
result, result - 1);
```

Output:

Round: 6, Win rate: 0.01505
 Round: 7, Win rate: 0.05433
 Round: 8, Win rate: 0.11383
 Round: 9, Win rate: 0.18978
 Round: 10, Win rate: 0.27061
 Round: 11, Win rate: 0.3585
 Round: 12, Win rate: 0.43604
 Round: 13, Win rate: 0.51252
 Ab 13 Runden hat der Spieler einen Vorteil.
 Die Bank darf demnach maximal $n = 12$ wählen.

Aufgabe 5**a)**

```

fn get_paths_a(
  n: u8, m: u8, path: &mut Vec<u8>, paths: &mut Vec<Vec<u8>>
) {
  if n == 0 && m == 0 {
    paths.push(path.clone());
  } else {
    if n > 0 {
      path.push(0);
      get_paths_a(n - 1, m, path, paths);
      path.pop();
    }
    if m > 0 {
      path.push(1);
      get_paths_a(n, m - 1, path, paths);
      path.pop();
    }
  }
}

```

b)

```

fn get_paths_b(
  n: u8, m: u8, c: u8, path: &mut Vec<u8>, paths: &mut Vec<Vec<u8>>
) {
  if n == 0 && m == 0 {
    paths.push(path.clone());
  }
}

```

```

    } else {
        if n > 0 {
            path.push(0);
            get_paths_b(n - 1, m, c + 1, path, paths);
            path.pop();
        }
        if m > 0 && c > 0 {
            path.push(1);
            get_paths_b(n, m - 1, c - 1, path, paths);
            path.pop();
        }
    }
}

```