

## 6. Programmieraufgabe Computerorientierte Mathematik II

**Abgabe: 12.6.2020** über den ComaJudge bis 18 Uhr

**Definitionen** Sei  $P_n \subset \mathbb{R}^2$  ein reguläres  $n$ -Eck mit Eckenmenge  $V_n = \{0, \dots, n-1\}$ , siehe Abbildung 1. Eine *Triangulierung* von  $P_n$  ist eine endliche Menge  $\Delta$  von Dreiecken mit folgenden Eigenschaften:

- Die Vereinigung über  $\Delta$  entspricht  $P_n$ .
- Die Eckenmengen der Dreiecke aus  $\Delta$  sind Teilmengen von  $V_n$ .
- Zwei Dreiecke aus  $\Delta$  schneiden sich in einer gemeinsamen Seite, in einem Punkt oder sind disjunkt. Im ersten Fall sind die beiden Dreiecke *adjacent* und der Schnitt ist ihre *Mauer*.

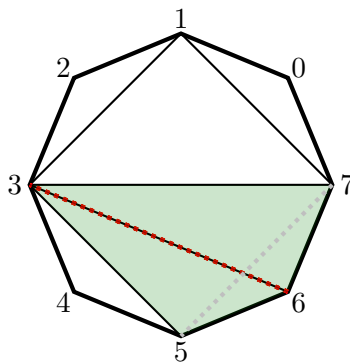


Abbildung 1: Eine Triangulierung des 8-Ecks mit 6 Dreiecken und 5 Mauern

- Ein *Flip* an einer Mauer  $M$  ( $[3,6]$ ) ist eine lokale Änderung der Triangulierung:  $M$  wird durch die andere Diagonale ( $[5,7]$ ) des umliegenden Vierecks ( $[3,5,6,7]$ ) ersetzt.

**Spezifikationen** Schreiben Sie eine Klasse `ngonTriang` mit den Attributen `n`, `triangles` und `walls` (wie üblich lexikographisch aufsteigend sortiert, s.u.), welche die folgenden Methoden bereitstellt:

- 1) Einen Konstruktor `__init__(self, n, triangles)`; Hierbei ist `triangles` eine Liste von dreielementigen Listen natürlicher Zahlen und  $n \geq 4$ . Der Konstruktor soll die Exception

`ValueError("no triangulation")`

werfen, falls die Eingabe keine Triangulierung beschreibt.

- 2) Eine Methode `n_walls(self)`, welche die Anzahl der Mauern zurückgibt.

3) Eine Methode `flip(self, wall)`, welche den Flip als `ngonTriang` - Objekt zurückgibt.

*Tipp (optional):* Implementieren Sie eine Funktion `_is_triangulation(walls)`.

### Beispielaufufe

```
1>>> n=4
2>>> triangles = [[0,1,2],[0,2,3]]
3>>> T = ngonTriang(n,triangles)
4>>> T.triangles
5 [[0, 1, 2], [0, 2, 3]]
6>>> T.walls
7 [[0, 2]]
8>>> T.n_walls()
9 1
10>>> S = T.flip([0,2])
11>>> S.triangles
12 [[0, 1, 3], [1, 2, 3]]
13>>> W = ngonTriang(n,[[0,1,2],[0,2,3],[1,2,3]])
14 Traceback (most recent call last):
15 ...
16 ...
17     raise ValueError("no triangulation");
18 ValueError: no triangulation
```

*Sie können diese Aufgabe bis einschließlich Montag, den 22.6.2020, bei Ihrer Tutorin oder Ihrem Tutor vorstellen.*