3.3.6 Lösung II: Triangulierungsverfeinerungsmeth.

Geg.: Planare Unterteilung G der Ebene, wobei G eine Triangulierung ist und auch der Rand von G (also die konvexe Hülle) ein Dreieck ist. $|CH| = 3$

- **Ziel:**
  Folge $S_1, S_2, \dots, S_h$ (mit $n_1, n_2, \dots, n_h$ Pktn) von Triangulierungen konstruieren, sodass gilt:
  1.) • $S_1 = G$ dh $n_1 = n = \#$ aller Pkte
  2.) • S_h besteht aus einem Dreieck, nämlich dem äußeren von G, d.h. $n_h = 3$
  3.) • n_{i+1} ist Bruchteil von n_i, dh $n_{i+1} \leq c \cdot n_i$ $c = const$ $c \leq 1$
  $\Rightarrow S_{i+1}$ besteht aus konstantem Bruchteil der Knoten von S_i
  4.) • Jeder Pkt q der in S_{i+1} lokalisiert ist, kann in $O(1)$ dh in konster Zeit in S_i lokalisiert werden.
  5.) • $S_1, \dots, S_h$ brauchen insgesamt Platz $O(n)$ folgt aus 3.)
  6.) • $h = O(\log n)$ folgt aus 3.)

- **Lösen des Point Location Problems:**
  - Geg sei Folge $S_1, \dots, S_h$
  - Lokalisation von q in S_h in Zeit $O(1)$
  - Unter Benutzung von 4.) lokalisieren wir nacheinander q in $S_{h-1}, S_{h-2}, \dots, S_1 = G$ jeweils in Zeit $O(1)$

  Da $h = O(\log n)$ haben wir insgesamt:
  - Suchzeit : $O(\log n)$
  - Platz : $O(n)$ folgt aus 5.) bzw 3.)

  } optimal !!

- **Definition:**
  Eine Teilmenge der Knotenmenge heißt unabhängige Knotenmenge (independent set), wenn in ihr keine 2 Knoten existieren die durch eine gemeinsame Kante verbunden sind.
  Beim Übergang von S_i zu S_{i+1} suchen wir max. Teilmenge der Knotenmenge sodass die angegebene Eigenschaft erfüllt ist.
  Um Eigenschaft 4.) zu erhalten sollten wir aber Knoten von möglichst kleinem Grad entfernen, und davon möglichst viele wegen Eigenschaft 3.).

- **Lemma 1:**
  Für jede Triangulierung G gilt:
  G enthält eine unabhängige Knotenmenge J sodass gilt:
  - $|J| \geq \left\lceil \frac{1}{12}\left(\frac{n}{2} - 3\right) \right\rceil$
  - $\forall v \in J$ gilt: $\text{degree}(v) \leq 11$
  - J enthält keinen der 3 Hüllknoten
  - J kann in Zeit $O(n)$ gefunden werden.

  Beweis:
  Betrachte folgenden Algorithmus:

  ```
  Markiere die 3 Hüllknoten
  J ← ∅
  repeat
      wähle Knoten v mit Grad ≤ 11 der nicht markiert ist
      J ← J ∪ {v}
      markiere v und alle seine Nachbarn
  until (∄ unmarkierten Knoten mit Grad ≤ 11)
  ```

  Der Algorithmus benötigt Zeit $O(n)$ und findet eine unabhängige Knotenmenge J von nicht Hüllknoten in der jeder Knoten höchstens Grad ≤ 11 hat.