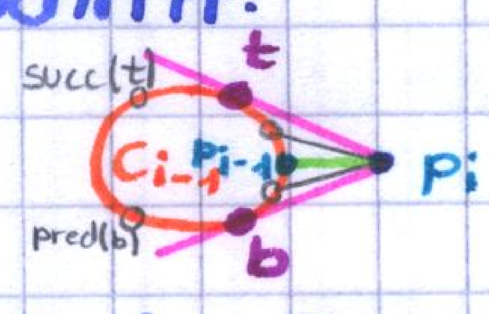
- Varianten von Graham's Scan:
  - ~~Aufgabe 1.2~~
  - Obere und untere Hülle gleichzeitig mit 2 Stacks
  - Gesamte Hülle in einer Phase
    - Sortiere $S = p_1, \dots, p_n$
    - Bearbeite p_i $i = 4, \dots, n$
      wobei $C_{i-1} = CH(\{p_1, \dots, p_{i-1}\})$

      p_{i-1} ist rechteste (obere) Ecke von C_{i-1}

      Schritt:

      Wir laufen von p_{i-1} nach oben bzw unten solange bis wir die Berührpkte t und b der beiden Tangenten von p_i an C_{i-1} gefunden haben.

      ```
      t ← p_{i-1}
      while (p_i, t, succ(t)) nicht left-turn do
          t ← succ(t)   Nachfolger
      od
      b ← p_{i-1}
      while (p_i, b, pred(b)) nicht right-turn do
          b ← pred(b)   Vorgänger
      od
      ```

      Entferne alle Pkte zwischen t und b und füge p_i nach b ein.

      Laufzeit:
      $\underbrace{O(1)}_{\text{für b und t}} + O(\#\text{ gestr. Pkte zw. b und t})$!? insgesamt: $O(n)$ ohne Sort, $O(n \log n)$ mit Sort

1.1.3 Triangulierung

- Definition:
  Die Triangulierung einer Pktmenge S ist eine Zerlegung der konvexen Hülle von S in disjunkte Dreiecke, sodass die Ecken der Dreiecke genau die Pkte in S sind.
  Es entsteht also ein planarer Graph $G = (V, E)$ mit $V = S$.

- Satz:
  Sei $|S| = n$, $\#$ Ecken von $CH(S) = h$
  Dann gilt:
  Jede Triangulierung von S besteht aus:
  - $3n - 3 - h$ Kanten
  - $2n - 2 - h$ Gebieten (dh Dreiecken)

- Algorithmus zur Berechnung einer Triangulierung:
  Ähnlich zu oben dargestellter Variante von Graham's Scan
  - Sortiere $S = p_1, \dots, p_n$
  - Start-Triangulierung T von p_1, p_2, p_3
  - for $i = 4, \dots, n$ berechne T_i
    ziehe von p_i Tangenten an T_{i-1} und trianguliere dazwischen.

- Laufzeit:
  - $O(n)$ ohne Sortieren
  - $O(n \log n)$ mit Sortieren