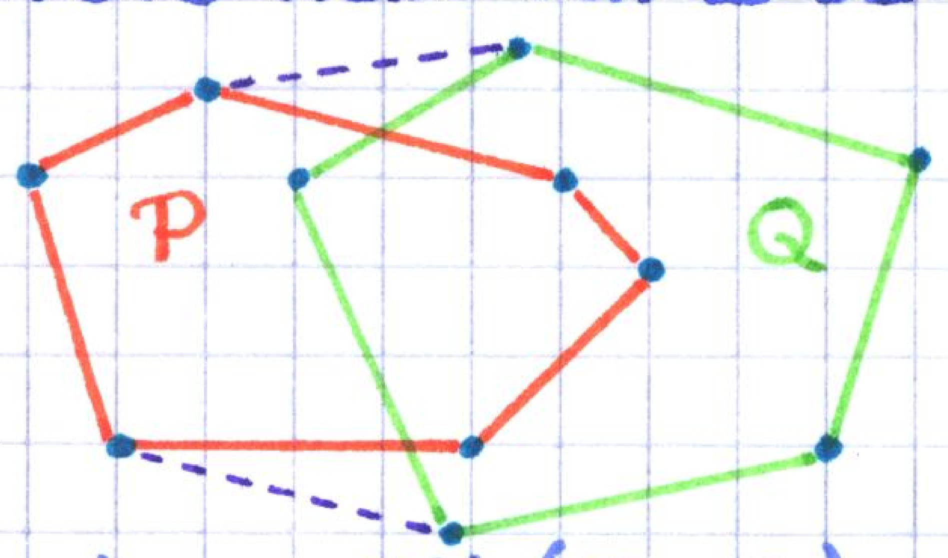
1.1.4 Algorithmus III: Divide & Conquer

Spezialfall:
Konvexe Hülle von 2 bel. konvexen Polygonen P und Q



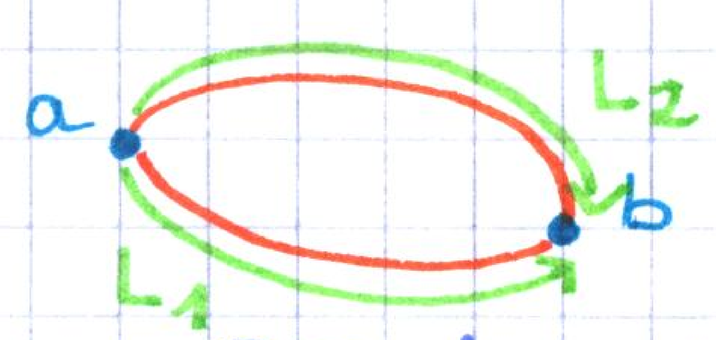
$P = p_1, \dots, p_m$
$Q = q_1, \dots, q_\ell$

Ecken gegen UZS sortiert

$S \subseteq \mathbb{R}^n$, $|S| = n = m + \ell$

Berechne CH(P ∪ Q)

- **Triviale Lsg.:**
  Graham's Scan auf der Menge aller n Pkte
  ⇒ Laufzeit $O(n \log n)$ mit Sortieren
- **Besser: Ausnutzen der Polygonstruktur:**
  Wir finden die Sortierung aller Ecken in Zeit $O(n)$ indem wir die Polygonstruktur ausnutzen
  ⇒ Laufzeit $O(n)$ mit Graham's Scan
  - Betrachte P:

    - Berechne extreme Ecken a und b in Zeit $O(m)$
      Lineare Suche auf einer m-elementigen Menge
    - L_1 = unterer Polygonzug
      Starte bei a, laufe gegen UZS über P bis b erreicht
      L_2 = oberer Polygonzug
      Starte bei b, laufe im UZS über P bis a erreicht, drehe um
    - Mische L_1 und L_2 zu einer sortierten Gesamtliste L_p
      zs (siehe Mischschritt von Mergesort) in Zeit $O(m)$
  - Betrachte Q: (geht analog)
    Sortierte Liste L_q der Ecken von Q in Zeit $O(\ell)$

  Mische L_p und L_q in Zeit $O(m+\ell) = O(n)$ zu sortierter Gesamtfolge zusammen.
- **Divide & Conquer - Algorithmus:**

```
CONVEX_HULL(S)
if |S| = 1 then
    output S
else
    · teile S in 2 möglichst gleich große Teile S1 und S2      } Divide
      (|S1| = ⌈|S|/2⌉ , |S2| = ⌊|S|/2⌋)
    · P ← CONVEX_HULL(S1)                                      } Conquer
      Q ← CONVEX_HULL(S2)                                        (rekursiver Aufruf)
    · Berechne CH(P ∪ Q) mit Graham's Scan                     } Misch-Schritt
end if
```

Laufzeit:
Teilen: $O(n)$
Mischen: $O(n)$

$$T(n) \leq \begin{cases} c_0 & ; n = 1 \\ c_1 \cdot n + 2 \cdot T(n/2) & ; n > 1 \end{cases}$$

wie bei Mergesort, $T(n) = \Theta(n \cdot \log n)$

⇒ insgesamt: $\Theta(n \cdot \log n)$