Algorithms & Probability PVK

Day 5

Michelle Honegger - CAB G59

Smallest Enclosing Circle

- **Given**: finite set of points $P \subseteq \mathbb{R}^2$
- **Seeked**: Circle with smallest radius, which encloses P
- For every point set $P \subseteq \mathbb{R}^2$. $|P| \geq 3$ there exists a subset $Q \subseteq P$, st. $|Q| = 3 \land C(Q) = C(P)$

Smallest Enclosing Circle

Algorithm

RANDOMISED_PRIMITIVEVERSION(P)

repeat forever

  wähle $Q \subseteq P$ mit $|Q| = 3$ zufällig und gleichverteilt

  bestimme $C(Q)$

  if $P \subseteq C^{\bullet}(Q)$ **then**

    return $C(Q)$

- Runtime: $T \propto Geo(1/\binom{n}{3}) \Rightarrow O(n \cdot n^3)$

Smallest Enclosing Circle

Algorithm

```
RANDOMISED_CLEVERVERSION(P)
repeat forever
    wähle Q ⊆ P mit |Q| = 11 zufällig und gleichverteil
    bestimme C(Q)
    if P ⊆ C•(Q) then
        return C(Q)
    verdoppele alle Punkte von P ausserhalb von C(Q)
```

- Runtime: $O(n \cdot T) = O(n \cdot log(n))$

Convex Hull

- **Given**: Pointset $P \subset \mathbb{R}^d$
- **Seeked**: convex hull of P, conv(P) as sequence $(q_1, q_2, \ldots, q_{h-1}), h \leq n$ vertices of the polygone starting with an arbitrary vertex, then counterclockwise
- **Convex hull**:= minimal convex set that contains P
- **Border Edge**:= ordered pair (q, r) for $q, r \in P$ with $q \neq r$ st. all dots of P are on the left side of (q, r)
- **General Position**:= no three dots are on one straight line and no 2 points have the same x-coordinate

Convex Hull

Jarvis Wrap

```
JarvisWrap(P)
h ← 0
p_now ← Punkt in P mit kleinster x-Koordinate
repeat
    q_h ← p_now
    p_now ← FindNext(q_h)
    h ← h + 1
until p_now = q_0
return (q_0, q_1, ..., q_{h-1})
```

- Runtime: O(n*h)
- Special Cases:
  - 2 points with smallest x-coordinate
    -> we choose the one with smallest y-coordinate
  - 3 or more points on one line
    -> We do $(det(p, q, q_{next}) < 0) \vee (det(p, q, q_{next}) = 0 \wedge |qp| > |qq_{next}|)$

Convex Hull

Local Repair

- **Improvement step**:= in case p,p′,p′′ are consecutive points and p′ is on the left of pp′′, we remove p′ out of the sequence
- **Runtime**: O(n*log(n))

```
LocalRepair(p_1, p_2, ..., p_n)
        ▷ setzt (p_1, p_2, ..., p_n), n ≥ 3, nach x-Koordinate sortiert, voraus
q_0 ← p_1
h ← 0
for i ← 2 to n do                        ▷ untere konvexe Hülle, links nach rechts
    while h > 0 und q_h links von q_{h−1}p_i do
        h ← h − 1
    h ← h + 1
    q_h ← p_i                  ▷ (q_0, ..., q_h) untere konvexe Hülle von {p_1, ..., p_i}
h′ ← h
for i ← n − 1 downto 1 do               ▷ obere konvexe Hülle, rechts nach links
    while h > h′ und q_h links von q_{h−1}p_i do
        h ← h − 1
    h ← h + 1
    q_h ← p_i
return (q_0, q_1, ..., q_{h−1}) ▷ Ecken der konvexen Hülle, gg. Uhrzeigersinn
```

Convex Hull

Planar Graphs

- **Planar Graph**:= Edges cross, if at all, only in endpoints
- **Triangulation**:= no edge can be added without resulting in 2 crossing edges
- **#improvementsteps** = 2n-2-h
  We start with 2n-2 edges, in each step we reduce 2 edges to 1, h stay in the end
- **#triangles in the triangulation** = 2n-2-h
  We begin with 0 triangles, in each step we create 1
- **#edges in the triangulation** = 3n-3-h
  In the beginning we have n-1 edges, in each step + 1

(d) Seien P und Q zwei Mengen von Punkten. Dann gilt $Conv(P \cap Q) = Conv(P) \cap Conv(Q)$.

WAHR ☐ FALSCH ☐

Begründung/Gegenbeispiel: *(2 Punkte)*