

Übungen zur Vorlesung

Ausgewählte Kapitel der Algorithmik – Geometrische Approximationsalgorithmen

WS 21/22

Blatt 2

Aufgabe 2.1 (k-Enclosing Disk)

Entwickeln Sie einen (einfacheren) Algorithmus, der in erwarteter $O(n(n/k))$ Zeit eine 2-Approximation für k-enclosing disk berechnet. Hierzu betrachten den folgenden Algorithmus:

- (1) Für jeden Punkt der Eingabe, füge ihn unabhängig, zufällig in eine Menge C ein,
- (2) Für jeden Punkt $q \in C$ bestimme die Distanz r_q zum $k - 1$ -nächsten Nachbarn
- (3) Teste, ob das kleinste r_q eine 2-Approximation ist (unter Verwendung eines Algorithmus von der Vorlesung), sonst starte den Algorithmus neu.

Aufgabe 2.2 ([Heimaufgabe] Smallest Enclosing Disk)

Analysieren Sie die Laufzeit des folgenden Algorithmus zum Berechnen einer smallest enclosing disk (SED).

```

SED( $P = \{p_1, \dots, p_n\}$ )
  if  $n \leq 3$  then  $D_n \leftarrow$  smallest enclosing disk( $P$ )
  else
     $P \leftarrow$  RandomPermute( $P$ )
     $D_2 \leftarrow$  SED( $\{p_1, p_2\}$ )
    for  $i = 3$  to  $n$  do
      if  $p_i \in D_{i-1}$  then  $D_i \leftarrow D_{i-1}$ 
      else  $D_i \leftarrow$  SEDwith1Point( $\{p_1, \dots, p_{i-1}\}, p_i$ )
  return  $D_n$ 

```

```

SEDwith1Point( $P = \{p_1, \dots, p_{i-1}\}, q$ )
   $P \leftarrow$  RandomPermute( $P$ )
   $D_1 \leftarrow$  SED( $\{p_1, q\}$ )
  for  $j = 2$  to  $i - 1$  do
    if  $p_j \in D_{j-1}$  then  $D_j \leftarrow D_{j-1}$ 
    else  $D_j \leftarrow$  SEDwith2Points( $\{p_1, \dots, p_{j-1}\}, p_j, q$ )
  return  $D_{i-1}$ 

```

```
SEDwith2Points( $P = \{p_1, \dots, p_{j-1}\}, q_1, q_2$ )
   $D_0 \leftarrow \text{SED}(\{q_1, q_2\})$ 
  for  $k = 1$  to  $j - 1$  do
    if  $p_k \in D_{k-1}$  then  $D_k \leftarrow D_{k-1}$ 
    else  $D_k \leftarrow \text{SED}(q_1, q_2, p_k)$ 
  return  $D_{j-1}$ 
```