

Triangulierung eines y -monotonen Polygons

Wolfgang Mulzer

Die Eingabe ist ein einfaches y -monotnes Polygon P in allgemeiner Lage, d.h., die y -Koordinaten aller Ecken von P sind paarweise verschieden. Die Ausgabe besteht aus einer Menge von Diagonalen, die P triangulieren. Der Algorithmus arbeitet folgendermaßen.

```
Erstelle eine Liste aller Ecken, von oben nach unten nach
y-Koordinate sortiert (u[1], u[2], u[3], ... , u[n]).

S <- new Stack()
// S.top() oberstes Element auf dem Stack
// S.secondTop() zweitoberstes Element auf dem Stack
S.push(u[1]); S.push(u[2])

for i := 3 to n-1 do
  if u[i] und S.top() sind auf verschiedenen Seiten von P then
    while S.size() > 1 do
      u <- S.pop()
      Erzeuge eine Diagonale zwischen u[i] und u
      S.pop(); S.push(u[i-1]); S.push(u[i]);
    else
      while S.size() > 1 AND
        die Kante u[i]-S.secondTop() liegt unter
        der Dreierkette u[i]-S.top()-S.secondTop() do
        S.pop()
        Erzeuge eine Diagonale zwischen u[i] und S.top()
        S.push(u[i])
      S.pop()
    while S.size() > 1 do
      u <- S.pop()
      Erzeuge eine Diagonale zwischen u[n] und u
```

Die Laufzeit für jeden Durchlauf der `for`-Schleife ist $O(1 + \text{Anzahl der } S.\text{pop() Operationen})$. In jedem Durchlauf der `for`-Schleife werden nur konstant viele Elemente auf S gepusht. Jedes Element, das gepopt wird, wurde vorher gepusht. Also ist die Gesamtlaufzeit $O(n)$.