

### → 3.4.5.13 Algorithmus für Point-Location:

Eingabe: Pkt  $q$

Datenstruktur  $D$ .

$D$ .search( $q$ ) liefert Dreieck von  $G$ , das  $q$  enthält.

if  $q$  außerhalb Dreieck i.d. Wurzel then

Aussage: "äußeres Gebiet von  $G$ ."

else

$v \leftarrow$  Wurzel

while  $v$  kein Blatt (d.h.  $D.outdeg(v) > 0$ ) do

forall Knoten  $u$  mit  $\exists$  Pointer  $v \rightarrow u$  do

if  $q$  innerhalb Dreieck von  $u$  then

$v \leftarrow u$ ;

fi

od

od

Ausgabe: Dreieck von  $v$

fi

Schleifeninvariante: Während der Ausführung der while-Schleife gilt stets, dass  $q \in$  Dreieck von  $v$ .

Bemerkung: Platz:

$$\begin{aligned} \# \text{Knoten} &= n_1 + n_2 + \dots + n_R \leq \\ &\leq n_1 + c \cdot n_1 + \dots + c^{R-1} \cdot n_1 = \\ &= n_1 \sum_{v=0}^{R-1} c^v = n_1 \cdot \frac{c^R - 1}{c - 1} = n \cdot \underbrace{\frac{c^R - 1}{c - 1}}_{\text{const}} = O(n) \end{aligned}$$