

Advanced Algorithms and Datastructures (Assignment 1)

Christian Müller, Daniel Kocher

March 6, 2016

input : Menge von horizontalen Segmenten H und eine Menge von vertikalen Segmenten V . Sei H^* die Menge der Endpunkte der horizontalen Segmente und $S = H^* \cup V$.

output: Paare (h, v) mit $h \in H$ und $v \in V$.

Sortiere Element von S gemäß ihrer horizontalen Position.;

ReportCuts(S, L_S, R_S, V_S);

Function *ReportCuts* (S, L_S, R_S, V_S)

```

    if |S| == 1 ;                                // Sei p dieses Element
    then
        begin
            switch p do
                case p ist linker Endpunkt
                    |  $L_S \leftarrow p_y, R_S \leftarrow \emptyset, V_S \leftarrow \emptyset$  return;
                end
                case p ist rechter Endpunkt
                    |  $L_S \leftarrow \emptyset, R_S \leftarrow p_y, V_S \leftarrow \emptyset$  return;
                end
                case p vertikales Segment
                    |  $L_S \leftarrow \emptyset, R_S \leftarrow \emptyset, V_S \leftarrow (p_{y1}, p_{y2})$  return;
                end
            endsw
        end
    else
        Wähle eine geeignete vertikale Trennlinie (dh. beliebige
        x-Koordinate), welche die Menge S in zwei gleich große
        Teilmengen  $S_1$  und  $S_2$  teilt;
        ReportCuts( $S_1, L_{S_1}, R_{S_1}, V_{S_1}$ );
        ReportCuts( $S_2, L_{S_2}, R_{S_2}, V_{S_2}$ );
         $tmp \leftarrow L_{S_1} \cap R_{S_2}$ ;
        /* tmp enthält so alle horizontalen Segmente, die in S1
           beginnen und in S2 enden. */
        print( $((L_{S_1} \setminus tmp) \odot V_{S_2})$ );
        /*  $(L_{S_1} \setminus tmp)$  enthält alle horizontalen Segmente, die in
           S1 beginnen und nicht in S2 enden.  $h \odot V_2$  liefert
           alle  $v \in V_{S_2}$ , die  $h_y$  enthalten. Genaue
           Vorgehensweise ist den Folien der Vorlesung zu
           entnehmen. */
        print( $((R_{S_2} \setminus tmp) \odot V_{S_1})$ );
        /* analog zu letztem print. Horizontale Segmente die
           in S2 enden und ihren Anfangspunkt nicht in S1
           haben geschnitten mit den vertikalen Segmenten aus
           S1. */
         $L_S \leftarrow (L_{S_1} \setminus tmp) \cup L_{S_2}$ ;
         $R_S \leftarrow (R_{S_2} \setminus tmp) \cup R_{S_1}$ ;
         $V_S \leftarrow V_{S_1} \cup V_{S_2}$ ;
        return;
    end
end

```

Algorithm 2: CutReporter