Blatt 3, Aufgabe 2: Lösungsvorschlag

Algorithmus. Jeder Punkt p_i wird mit einem vertikalen Strahl versehen, der bei p_i startet und nach oben zeigt. Zum Erstellen des geometrischen Graphen sweept der Algorithmus in x-Richtung über die Punkte. Der Sweep-Line-Status ist ein Stack, der zu jedem Zeitpunkt die vertikalen Kanten (absteigend nach y-Koordinaten sortiert) hält, die nicht nach rechts von anderen Kanten verdeckt sind. Dabei bezeichnen wir eine Kante als *relevant*, wenn ihr unterer Knoten einer der zuvor betrachteten p_i ist. Alle anderen Kanten sind *irrelevant*.

An einem Eventpunkt p_i werden alle Kanten e, deren unterer Punkt p_e oberhalb von p_i liegt, abgelaufen und entfernt. Ist eine dabei betrachtete Kante e relevant, wird von p_e eine horizontale Kante zum Strahl von p_i eingefügt. Dabei entsteht auf dem Strahl von p_i eine neue vertikale Kante. Ist der Stack danach leer (p_i ist der bisher tiefste Punkte) wird eine Kante von p_i nach links ins Unendliche eingefügt. Andernfalls wird die oberste Stack-Kante e vom Stack entfernt und eine horizontale Kante zwischen p_i und e erstellt. Die Kante e wird dabei in zwei vertikale Teile aufgeteilt, von denen der untere Teil auf den Stack gelegt wird. Alle Kanten oberhalb von p_i wurden aus dem Stack entfernt, weil sie nun nach rechts von neuen Kanten (auf dem Strahl von p_i) verdeckt werden. Diese neuen Kanten werden auf den Stack gelegt und als irrelevant markiert. Lediglich die unterste Kante, die p_i selbst als unteren Punkt hat, wird als relevant markiert. Ganz am Ende müssen die unteren Punkte aller relevanten Kante, die noch auf dem Stack liegen, nach rechts ins Unendliche verbunden werden.

Laufzeit. Es wird einmal über alle n Punkte iteriert. Für jeden Punkt p_i werden genau zwei vertikale Kanten erstellt und auf den Stack gelegt: beim Verbinden nach links die Kante unterhalb von p_i, beim Verbinden nach rechts die Kante oberhalb von p_i. Die dabei durchgeführten Operationen (horizontale Kanten einfügen, vertikale Kanten ggfs. aufteilen und einfügen) brauchen $\mathcal{O}(1)$ Zeit. Damit ergibt sich amortisiert eine Laufzeit von $\mathcal{O}(n)$. Zum Schluss werden für alle relevanten Kanten, die noch auf dem Stack liegen, Kanten nach rechts ins Unendliche erstellt. Auch das geschieht in $\mathcal{O}(1)$ pro Kante und damit insgesamt in $\mathcal{O}(n)$.