Highest-Label-Implementation

$\boxed{\text{Wähle jeweils einen aktiven Knoten } v \text{ mit } \mathit{dist}(v) \text{ maximal.}}$

Für die effiziente Bestimmung des jeweils nächsten Knotens verwalte die aktiven Knoten in Listen (*buckets*) b_i, $i = 0, \ldots, 2n-1$, und zwar so, dass

$$v \in b_i \iff v \text{ aktiv und } \mathit{dist}(v) = i$$

gilt. Beachte, dass immer $0 \leq \mathit{dist}(v) \leq 2n-1$ (Lemma 11).

Ein Index $d \in \{0, \ldots, 2n-1\}$ zeigt auf die nicht-leere Liste mit dem jeweils höchsten Index und wird wie folgt aktualisiert:

- nach RELABEL(v): $d \leftarrow \max\{d, \mathit{dist}(v)\}$
- nach PUSH(v, w): **while** b_d *leer* **do** $d \leftarrow d - 1$

Der jeweils nächste Knoten kann dann beliebig aus b_d gewählt werden.

Lemma: Die Gesamtlaufzeit aller Aktualisierungen ist in $\mathcal{O}(n^2)$.

Beweis: Wegen Lemma 11 sind sowohl die Anzahl der RELABEL-Aufrufe als auch die Summe der dabei gemachten Erhöhungen in $\mathcal{O}(n^2)$. Dann kann aber auch die Anzahl der Erniedrigungen von d nach PUSH-Aufrufen nicht größer sein. □

Zu jedem Knoten $v \in V \setminus \{t\}$, für den bei vorhandenem Überschuss ein PUSH zulässig wäre, sei die *PUSH-Kante* $p(v) = (v, w)$ diejenige Kante, welche unsere Implementation für die nächste PUSH-Operation auswählen würde. Wir nehmen an, dass sich $p(v)$ nur ändert, wenn die Kante keine Restkapazität mehr hat.

Die PUSH-Kanten induzieren einen Wald F, denn es gibt höchstens $n-1$ viele und der induzierte Teilgraph kann keinen Kreis enthalten, da $\mathit{dist}(v) < \mathit{dist}(w)$ für jede PUSH-Kante (v, w). Daraus folgt auch, dass die Bäume in F Wurzelbäume sind, in denen alle Kanten in Richtung der Wurzel zeigen. Für $v \in V$ sei $D(v)$ die Menge der Nachfolger von v im zugehörigen Baum. Weil jeder Knoten zumindest sich selbst als Nachfolger hat, gilt $|D(v)| \geq 1$.

Ein aktiver Knoten heißt *maximal*, wenn er keinen aktiven Nachfolger in F hat. Bezeichne H die Menge der maximalen aktiven Knoten. Für $v, w \in H$ gilt $D(v) \cap D(w) = \emptyset$, denn sonst wäre einer Nachfolger des anderen und damit nicht maximal.

Um zu zeigen, dass die Anzahl der nicht-saturierenden PUSH-Operationen in $\mathcal{O}(n^2\sqrt{m})$ ist, werden wir eine amortisierte Analyse mit der Potenzialmethode durchführen. Für eine noch festzulegende Konstante K sei das Potenzial des Waldes F definiert durch

$$\Phi = \Phi(F, H) = \sum_{v \in H} \Phi(v) \quad \text{mit} \quad \Phi(v) = \max\{0, K + 1 - |D(v)|\} \ .$$

Beachte, dass $0 \leq \Phi(v) \leq K$. Das Potenzial kann sich nur ändern, wenn H oder F sich ändern, also bei einem PUSH oder RELABEL. Wir unterscheiden die folgenden Fälle.

Lemma: Für einen maximalen aktiven Knoten $v \in H$ mit $p(v) = (v, w)$ (falls ex.) gilt:

(i) Durch ein nicht-saturierendes PUSH(v, w) wird Φ nicht erhöht, aber um mindestens 1 verringert, falls $|D(v)| \leq K$.

(ii) Durch ein saturierendes PUSH(v, w) wird Φ um höchstens K erhöht.

(iii) Durch ein RELABEL(v) wird Φ um höchstens K erhöht.

Beweis: Wir stellen zunächst fest, dass am Ende der Operationen einzufügende PUSH-Kanten das Potenzial nicht erhöhen, weil dadurch die Menge H der maximalen aktiven Knoten höchstens verkleinert und die Zahl der Nachfolger jedes Knotens höchstens vergrößert wird.

Bei einem nicht-saturierenden PUSH(v, w) wird der ganze Überschuss von v nach w geleitet, ohne den Wald zu verändern. Knoten w kann dadurch ein maximaler aktiver Knoten werden. Weil aber $|D(v)| < |D(w)|$ gilt, wird Φ um mindestens eine Einheit kleiner, falls $|D(v)| \leq K$, und bleibt ansonsten unverändert.

Nach einem saturierenden PUSH(v, w) ist (v, w) nicht mehr im Wald F, wodurch w ein maximaler aktiver Knoten werden und das Potenzial sich um bis zu K erhöhen kann.

Wird ein RELABEL(v) ausgeführt, hat v keine für PUSH zulässige ausgehende Kante, muss als Wurzel eines Baumes in F sein. Von den eingehenden PUSH-Kanten ist nach RELABEL keine mehr zulässig, sodass die Anzahl der Nachfolger sich auf einen (v selbst) verringert. Das Potenzial an v wird dadurch um höchstens K erhöht, und weil den vorherigen Nachfolgern keiner aktiv gewesen sein kann ($v \in H$), kommt kein maximaler aktiver Knoten neu hinzu. $\square$

Lemma: Es werden höchstens $\mathcal{O}(n^2\sqrt{m})$ nicht-saturierende PUSH-Operationen ausgeführt.

Beweis: Wir wissen bereits, dass höchstens $\mathcal{O}(n^2)$ RELABEL-Operationen vorkommen. Diese unterteilen die Folge aller PUSH-Operationen in $\mathcal{O}(n^2)$ dazwischen liegende *Phasen*, in denen sich die *dist*-Markierungen nicht ändern. Eine Phase ist *billig*, wenn darin höchstens $\frac{2n}{K}$ nicht-saturierende PUSHs vorkommen, und sonst *teuer*. In den billigen Phasen werden insgesamt höchstens $\mathcal{O}(n^2 \cdot \frac{2n}{K}) = \mathcal{O}(\frac{n^3}{K})$ nicht-saturierende PUSHs ausgeführt. Für die teuren Phasen benutzen wir das Potenzialargument.

In einer teuren Phase werden mindestens $\frac{2n}{K}$ nicht-saturierende PUSH-Operationen an maximalen aktiven Knoten ausgeführt. Davon werden wegen der disjunkten Nachfolgermengen weniger als $\frac{n}{K}$ viele an Knoten mit mehr als K Nachfolgern ausgeführt. Über alle Phasen aufsummiert sind dies wieder $\mathcal{O}(\frac{n^3}{K})$. Die übrigen verringern Φ nach dem vorstehenden Lemma um jeweils mindestens 1.

Andererseits gibt es höchsten $\mathcal{O}(nm)$ viele RELABEL- oder saturierende PUSH-Operationen, sodass die Summe aller Erhöhungen durch $\mathcal{O}(nmK)$ beschränkt ist. Es gibt also auch nicht mehr als $\mathcal{O}(nmK)$ nicht-saturierende PUSHs, die Φ verringern.

Wählen wir nun $K = \frac{n}{\sqrt{m}}$, dann sind alle diese Anzahlen in $\mathcal{O}(n^2\sqrt{m})$. $\square$

Da die Anzahlen der RELABEL- und saturierenden PUSH-Operationen immer durch $\mathcal{O}(n^2)$ bzw. $\mathcal{O}(nm)$ beschränkt sind, erhalten wir wegen $m \in \mathcal{O}(n^2)$ das gewünschte Ergebnis.

Satz: Die Highest-Label-Implementation benötigt $\mathcal{O}(n^2\sqrt{m})$ PUSH- und RELABEL-Operationen.

Bemerkung: Die Operationen und Auswahlen können mit konstanter amortisierter Laufzeit implementiert werden.