Dr. Andreas Borchert
Christian Ehrhardt
Institut für Angewandte Informationsverarbeitung

Allgemeine Informatik III (WS 2006/2007)

Abgabetermin 12.Dezember 2006

Super- und Subsequence

Wie bereits im letzten Blatt angedeutet, wird in diesem Blatt die Suche nach dem kürzesten Weg in einem Graphen auf ein anderes Problem angewandt. Unser eigentliches Ziel ist es unter anderem, zu zwei gegebenen Strings – etwa ANANAS und BANANE – den kürzesten String zu finden, der beide Strings enthält. Dabei heißt *enthalten* einfach, daß man den enthaltenen String durch das Streichen einiger Buchstaben bekommen kann. Für das Beispiel ANANAS und BANANE wäre z.B. BANANEAS ein solcher String. Auch ABNANANES würde beide Strings enthalten, ist allerdings länger als BANANEAS und damit nicht die gesuchte Lösung. Man sieht an diesem Beispiel auch gleich, daß ANANAS nicht "am Stück" vorkommt, das ist also durchaus erlaubt.

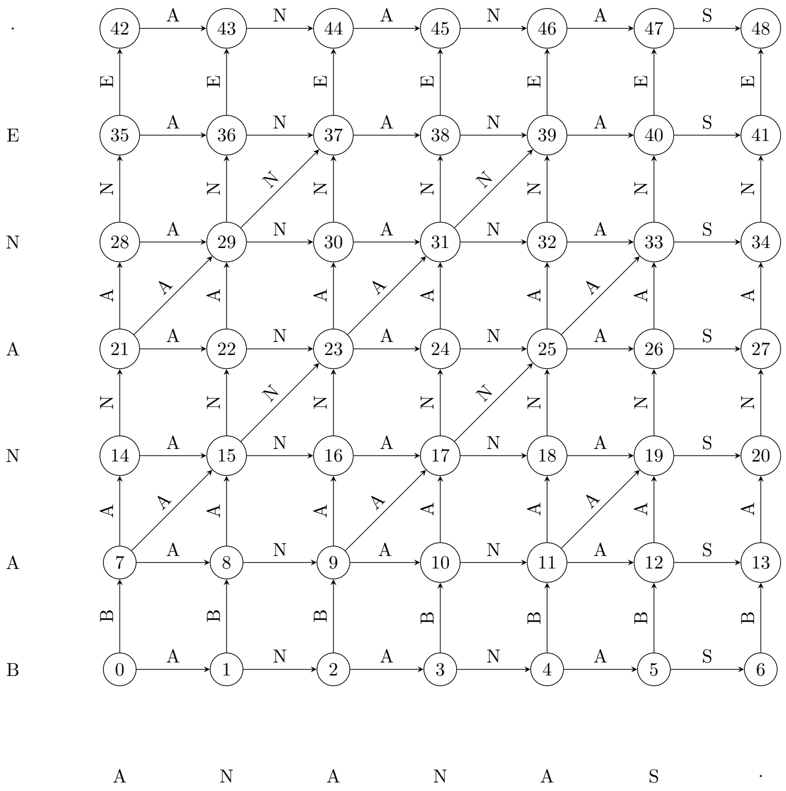
Aufgabe

In diesem Blatt soll ein Programm entwickelt werden, das zu zwei gegebenen Strings zwei Dinge bestimmt:

- Den kürzesten String, der beide gegebenen Strings enthält (Shortest Common Supersequence) und
- Den längsten String, der in beiden gegebenen Strings enthalten ist (Longest Common Subsequence). Bei ANANAS und BANANE wäre das ANAN).

Ein Graph

Was hat das nun mit kürzesten Wegen zu tun? Dazu betrachten wir das folgende Bild:
Man schreibt hier zunächst den einen String (im Bild ANANAS) waagerecht, den anderen String (BANANE) senkrecht. Dadurch entsteht eine Art rechteckige Tabelle, an die oben eine Zeile und rechts eine Spalte hinzugefügt werden muß. Die Zellen dieser Tabelle bilden die Knoten des Graphen und werden (z.B. zeilenweise) durchnumeriert. Im Bild ergibt das einen Graphen mit 49 Knoten.
Von jedem Knoten geht immer eine Kante zum direkt rechts und eine zum direkt oberhalb liegenden Knoten. Außerdem geht eine Kante zum schräg rechts oben liegenden Knoten genau



dann, wenn der zur Zeile und der zur Spalte gehörende Buchstabe gleich sind. Die Kanten sind hier immer mit dem Buchstaben beschriftet, dessen Zeile und/oder Spalte in der Tabelle entlang des Pfeils verlassen wird. In dem so entstandenen Graphen hat also jeder Knoten höchstens drei Nachbarn.

Wege

Betrachtet man irgendeinen Weg von links unten (Knoten 0) nach rechts oben (Knoten 48), dann kann man entlang der Kanten, die nach rechts (schräg oder geradeaus) gehen, immer das Wort ANANAS lesen. Genauso kann man, wenn man nur die Kanten betrachtet, die gerade oder schräg nach oben gehen immer das Wort BANANE lesen. Jeder Weg von links unten nach rechts oben enthält also sozusagen die Worte ANANAS und BANANE. Das heißt, daß jeder String, der sich entlang eines Weges von links unten nach rechts oben ergibt, sowohl das Wort ANANAS alsauch das Wort BANANE enthält.

Shortest Common Supersequence

Suchen wir also den kürzesten Weg von links unten nach rechts oben, so ergibt sich entlang dieses Wegs der kürzeste String, der beide gegebenen Strings enthält. Zur Suche nach dem kürzesten Weg kann auf der Lösung zum letzten Blatt aufgebaut werden.

Longest Common Subsequence

Entlang eines solchen Weges bilden genau die Buchstaben an den schräg verlaufenden Kanten den längsten String, der in beiden Strings enthalten ist.

Effizienz

Wenn die Strings länger werden, gilt es einige Dinge zu beachten, damit das Programm nicht zu lange läuft. Beide Punkte können aber zunächst hinten angestellt werden.

Auswahl der Knoten

Bereits im letzten Blatt wurde angedeutet, daß die Auswahl des jeweils nächsten Knotens einfach in der Reihenfolge, in der die Knoten entdeckt wurden, geschehen kann.

Repräsentation des Graphen

Da jeder Knoten in unserem Graphen nur höchstens 3 Nachbarn hat, lohnt es sich, den Graph nicht mehr in Form einer Matrix zu repräsentieren. Statt dessen merkt man sich einfach zu jedem Knoten, welche zwei bzw. drei Knoten über eine direkte Verbindung zu erreichen sind. Das spart nicht nur Platz sondern auch Zeit bei der Suche nach Nachbarn.

Hinweis

Für viele Beispiele gibt es mehrere korrekte Lösungen. Z.B. ist für die zwei Strings AA und AB sowohl AAB alsauch ABA kürzester String, der beide gegebenen Strings enthält. Wenn es mehrere längste bzw. kürzeste Lösungen gibt, spielt es keine Rolle, welche davon Euer Programm findet.

Viel Erfolg!