

Tutorium 12

Algorithmen I SS 14

Institut für Theoretische Informatik



- Global optimale Lösungen bestehen oft aus lokal optimalen Lösungen
- Wähle für die *aktuelle Situation* die optimale Entscheidung.
- Keine Reversion von vergangenen Entscheidungen
- \Rightarrow Einfach und schnell

Greedy Algorithmen die optimale Ergebnisse liefern:

- Dijkstra für kürzeste Wege
- Beide MST Algorithmen
 - Kruskal
 - Jarnik-Prim
- Selection Sort

Dynamische Programmierung

Definition

Dynamische Programmierung ist ein algorithmisches Muster, das vor allem zur Lösung von Optimierungsproblemen eingesetzt wird.

Idee:

- Problem in Teilprobleme zerlegen
- Lösungen der Teilprobleme speichern
- Endlösung aus Lösungen der Teilproblemen zusammensetzen

Dynamische Programmierung: Zwischenspeichern der Ergebnisse

- in einem ein- oder mehrdimensionalen Array
- Basisfälle vorinitialisieren
- Rest der Tabelle wird mit Werten für 'nicht berechnet' zB. -1 initialisiert
- für jeden Funktionsaufruf Table-Lookup:
 - steht ein Wert in der Tabelle: zurückgeben
 - sonst: Wert berechnen, in der Tabelle speichern und zurückgeben

- 'Vom Problem zur Lösung'
- rekursiver Funktionsaufruf
- overhead durch viele Funktionsaufrufe
- es kann bei extrem großen Problemen zu stackoverflow kommen

- die Tabelle wird sukzessiv mit Lösungen gefüllt
- das Endergebniss wird danach einfach aus der Tabelle abgelesen

Die sog. *Levenshtein Distanz* gibt eine Editierdistanz, also eine Art Abstand, zwischen zwei Wörtern an. Sie ist definiert als die minimale Anzahl an Einfüge-, Lösch- und Ersetzungsoperationen, um ein Wort in das andere umzuwandeln.

Lösung mit dynamischer Programmierung!

$$\text{lev}_{a,b}(i, j) = \min \begin{cases} \text{lev}_{a,b}(i-1, j) + 1 & (a \text{ einfügen}/b \text{ löschen}) \\ \text{lev}_{a,b}(i, j-1) + 1 & (a \text{ löschen}/b \text{ einfügen}) \\ \text{lev}_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} & (\text{ersetzen}) \end{cases}$$

Palindrome sind Wörter, die rückwärts gleich wie vorwärts sind (z.B. „anna“, „kajak“, „reittier“).

Aufgabe 1: Entwickle einen Algorithmus, der für eine Zeichenkette der Länge n in $\mathcal{O}(n^2)$ Zeit die Länge der längsten palindromischen Teilsequenz.

Beispiel: „FAKULTÄTSFEST INFORMATIK 2014“

Aufgabe 2: Erweitere den Algorithmus, sodass er zusätzlich diese Sequenz ausgibt.

Zwei Kohleminen müssen mit Essen versorgt werden. Es gibt drei verschiedene Arten von Essen: Fleisch, Fisch und Brot. Die Arbeiter mögen Abwechslung und sind produktiver, wenn sie verschiedene Arten von Essen bekommen. Für die nächste Produktion entscheidend sind immer die letzten drei Sendungen.

- Wenn alle Sendungen gleich sind \Rightarrow 1 Kohleereinheit
- Wenn es zwei verschiedene Sendungen gab \Rightarrow 2 Kohleereinheiten
- Wenn alle verschieden waren \Rightarrow 3 Kohleereinheiten

Für eine gegebene Reihenfolge an Essenssendungen kann jeweils entschieden werden, welche Mine sie erhält.

Aufgabe: Berechne die maximale Anzahl an produzierten Kohleereinheiten.