Institut für Informatik
Ludwig-Maximilians-Universität München

SS 2008
23.06.2008
Übungsblatt 11

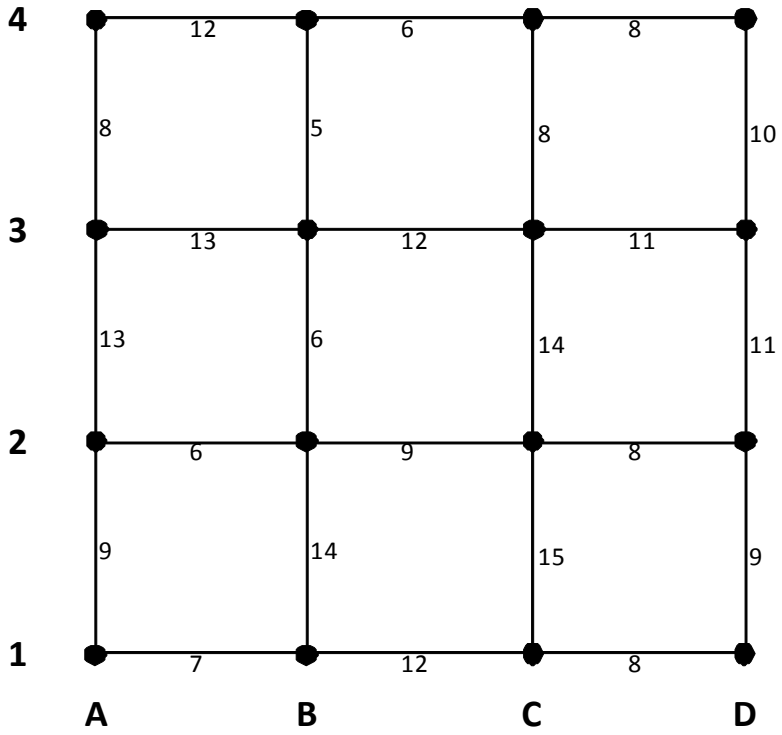
Prof. Dr. Hans-Peter Kriegel
Thomas Bernecker, Tobias Emrich

Übungen zur Vorlesung

Effiziente Algorithmen

Hausaufgabe 10.1: *Backtracking/ Branch-and-Bound* **(3 + 3 Punkte)**

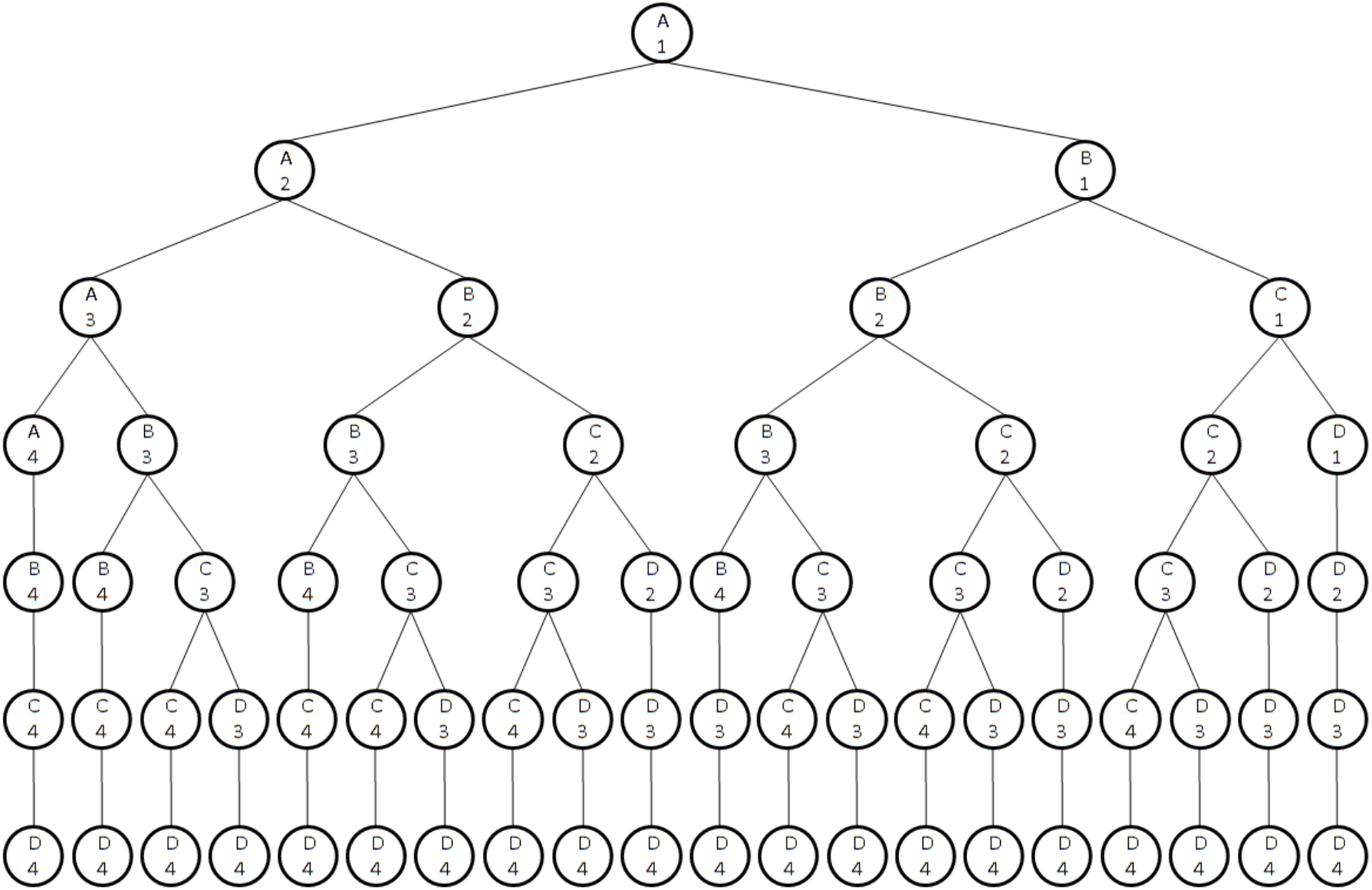
Gegeben sei folgendes Straßennetzwerk mit Kreuzungen (A1 - D4) und Kanten als Verbindungsstrecken zwischen diesen. Die Zahlen rechts und unterhalb der Kanten entsprechen der Fahrtdauer (in Minuten) zum Befahren der zugehörigen Straße. Aufgrund von Staus, Baustellen, Geschwindigkeitsbegrenzungen, usw. variieren die Fahrtzeiten zwischen 5 (schnellstmögliche Befahrung) und 15 Minuten.



Ziel ist es nun die schnellstmögliche Route zwischen A1 und D4 zu finden. Dazu müssen aufgrund der Gegebenheiten nur Wege betrachtet werden, die von jeder Kreuzung aus nach rechts bzw. oben fortfahren. Markieren Sie welche Teilbäume im Lösungsbaum nicht betrachtet werden, wenn zur Lösung des Problems ...

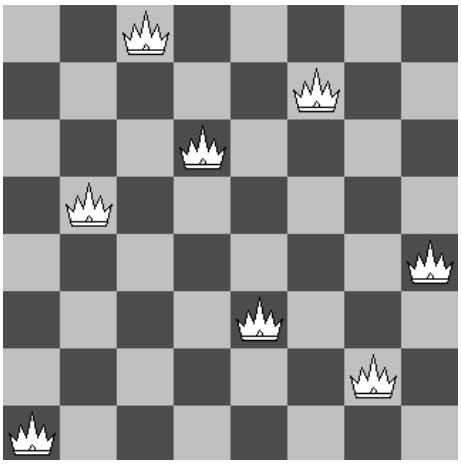
a) ... Backtracking verwendet wird.

b) ... Branch-and-Bound verwendet wird.


A 1
A 2
B 1
A 3
B 2
B 2
C 1
A 4
B 3
B 3
C 2
B 3
C 2
C 2
D 1
B 4
B 4
C 3
B 4
C 3
C 3
D 2
B 4
C 3
C 3
D 2
C 3
D 2
D 2
C 4
C 4
C 4
D 3
C 4
C 4
D 3
C 4
D 3
D 3
D 3
C 4
D 3
C 4
D 3
D 3
C 4
D 3
D 3
D 3
D 4
D 4
D 4
D 4
D 4
D 4
D 4
D 4
D 4
D 4
D 4
D 4
D 4
D 4
D 4
D 4
D 4
D 4
D 4
D 4

Hausaufgabe 10.2: *Acht-Damen-Problem* **(5 Punkte)**

Acht Damen sind so auf einem Schachbrett (acht Zeilen und acht Spalten) zu platzieren, dass keine zwei Damen sich gegenseitig schlagen können (siehe Abbildung). Zwei Damen können sich schlagen, wenn sie in der selben Zeile, Spalte oder Diagonalen stehen. Schreiben Sie ein Java-Programm, das in geeigneter Form eine mögliche Positionierung ausgibt. Bei der Lösung des Problems sollen Sie die in der Vorlesung besprochene Programmiertechnik „Backtracking“ anwenden.



Hausaufgabe 10.3: *Divide-and-Conquer* **(2 + 4 Punkte)**

Divide-and-Conquer-Methoden dienen dazu, Probleme durch deren rekursive Zerlegung in Teilprobleme schneller und damit effizienter zu lösen. Bei den Sortierverfahren wurden bereits einige konkrete Verfahren vorgestellt, die auf der Divide-and-Conquer-Methode basieren (z.B. Merge-Sort).

Nun soll eine ähnliche Ausgangslage betrachtet werden: Gegeben ist ein Array der Länge 2^n, das natürliche Zahlen enthält. Aus diesen Eintragen sollen Maximum und Minimum bestimmt werden.

a) Geben Sie einen einfachen Algorithmus (in Java-Notation) zur Bestimmung dieser Extrema an. Wieviele Vergleiche sind damit notwendig?

b) Geben Sie nun einen Algorithmus (ebenfalls in Java-Notation) an, der die Divide-and-Conquer-Methode anwendet. Erläutern Sie die Vorteile bzgl. der Vergleiche im Gegensatz zum naiven Ansatz aus Aufgabe a).