RWTH Aachen
Lehrgebiet Theoretische Informatik
Kuinke–Rossmanith–Sanchez

WS 2016
Lösungsvorschlag 9
21.12.2016

Übung zur Vorlesung Effiziente Algorithmen

Tutoraufgabe 24

Zunächst ist es problematisch, daß es unendlich viele Möglichkeiten für r_i gibt. Falls k vorgegeben ist, dann ist klar, daß zwei Zahlen äquivalent sind, falls sie in der gleichen Hälfte, im gleichen Drittel, und so weiter bis zum gleichen $k$tel sind. Wir können also einfach alle Zahlen i/j mit $2 \leq j \leq k$ und $0 \leq i \leq j$ betrachten und nur aus dieser Menge Kandidaten wählen. Jetzt können wir einfach Backtracking durchführen und eine neue Zahl verwerfen, wenn sie in einem der gemeinsamen Unterintervalle fällt. Ein Schritt im Backtracking läßt sich so in $O(k)$ Schritten implementieren.

Tutoraufgabe 25

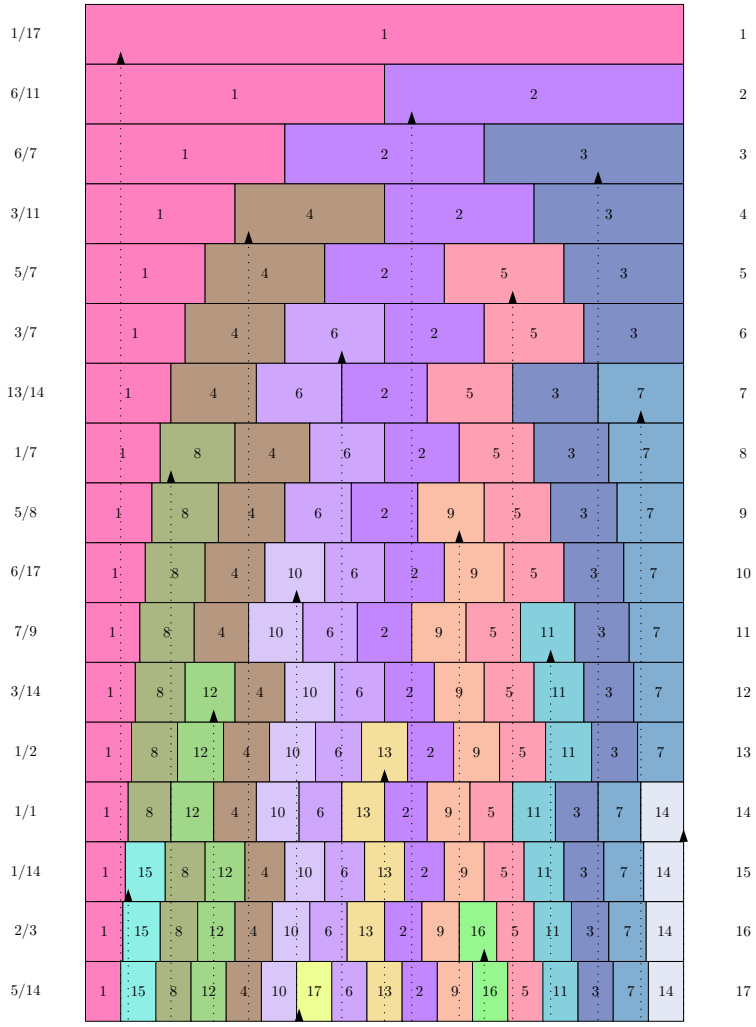
a) Wenn wir n Puzzleteile haben, dann gibt es $n!$ Platzierungen.

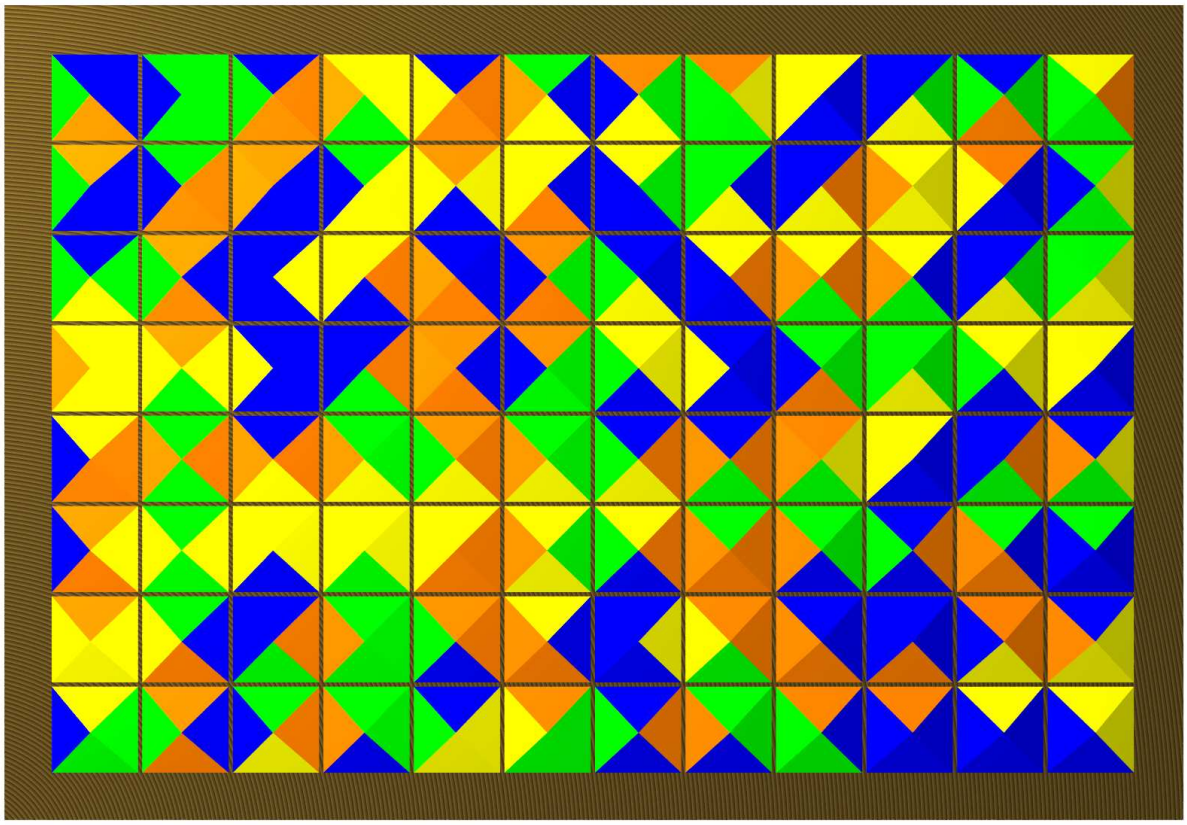
b) Mit dem Verfahren der Vorlesung können wir die Größe des Rekursionsbaums mithilfe Traversierung eines zufälligen Pfads schätzen.

c) Einfaches Backtracking ist wohl zu naiv, aber wenn man mit verschiedenen zufälligen Reihenfolgen beginnt und nach kurzer Zeit die Rekursion neu startet, hat man eine gute Chance schnell eine Lösung zu finden.

Hausaufgabe 16 (15 Punkte)

Es stellt sich heraus, daß es bis $k = 17$ Lösungen gibt und danach nicht mehr. Folgende Zeichnung zeigt die lexikographisch erste Lösung mit der Diskretisierungsstrategie aus der Tutoraufgabe.



Hausaufgabe 17 (15 Punkte)

Hier ist ein mögliche Lösung. Es gibt viele. Dies beantwortet eine offene Frage aus dem *Handbuch der Algorithmen.*