Professur für
Quantenalgorithmik

Universität Augsburg
Fakultät für Angewandte
Informatik

Informatik 3 WS 23/24

Übungsblatt 5 (KW 47)

5.1 Effizient Raten

In der Vorlesung hatten wir uns eine Definition der Komplexitätsklasse NP angeschaut, welche unabhängig vom Rechnermodell der nichtdeterministischen Turingmaschine war. An dieser Stelle wollen wir es kurz wiederholen: Ein Entscheidungsproblem A ist in NP, wenn es ein weiteres Entscheidungsproblem B gibt welches zwei Eingaben x und y verarbeitet und folgendes gilt

1. B kann von einer Turingmaschine in Polynomialzeit entschieden werden.
2. $A(x) = 1 \Leftrightarrow \exists y\ B(x, y) = 1$.
3. $A(x) = 0 \Leftrightarrow \forall y\ B(x, y) = 0$.
4. $|y| \in O(p(|x|))$.

Hierbei steht $|x|$ für die Länge der Eingabe x (bzw. y) und $p(|x|)$ für ein beliebiges Polynom von $|x|$. Zeigen Sie: Wenn wir unter 4. strikter beschränken, z.B. mit $|y| \in O(\log(|x|))$, dann können wir das Entscheidungsproblem A in Polynomialzeit lösen.
Tipp: Sie können gern von einer Kodierung in Bits ausgehen.

5.2 Konkret Raten

Zeigen Sie: Das Entscheidungsproblem zum 0/1-Rucksackproblem liegt in NP. Zur Erinnerung: Das Entscheidungsproblem ist: “Gibt es zur gegebenen Grundmenge U einen Rucksack mit Maximalgewicht M der mindestens den Wert X erreicht”.

5.3 InsertionSort

In der Vorlesungen haben wir die Grundidee von `InsertionSort` kennengelernt und die Laufzeit analysiert. Geben Sie ein explizites Programm für `InsertionSort` in einer Sprache Ihrer Wahl an – bei der Umsetzung der “FIND(B,j)” Subroutine haben Sie die Wahl. Testen Sie Ihr Programm und analysieren Sie die Laufzeit (best-case, und worst-case).

Falls Sie Spaß an Templates oder Duck-Typing haben, versuchen Sie gerne einen allgemeine Implementierung. Ansonsten reicht es aus wenn Sie sich auf Arrays mit ganzen Zahlen als Eingabe beschränken.