# Übungen zur Vorlesung Quantencomputer – Blatt 8
# Wintersemester 2011*

(Dated: 16. Dezember 2011 – Abgabetermin Freitag, 13. Januar 2012, 10:00 Uhr)

## I. GROVER ALGORITHMUS

Um uns die Funktionsweise des Grover Algorithmus noch etwas klarer zu machen, wollen wir ihn auf dem Computer simulieren. Schreibe dazu ein Programm, dass für eine Datenbank mit $N = 2^n$ ($n$ möglichst beliebig) Einträgen den Grover Algorithmus simuliert. Es gebe $M$ Lösungen des Orakels, d.h. genau für alle $|x\rangle \in \mathbf{M}$ mit $M = |\mathbf{M}|$ gilt $\hat{O}|x\rangle = -|x\rangle$, sonst $\hat{O}|y\rangle = |y\rangle$.

Deine Lösung sollte beispielhaft die folgenden Ergebnisse dokumentieren:

1. Die Matrix des Orakeloperators $O$ für $n = 3$ und irgendwelche $M = 2$ Lösungen (1 Punkt).
2. Die Matrix des Groveroperators für $n = 3$ und irgendwelche $M = 2$ Lösungen (1 Punkt).
3. Das Ergebnis $|\psi_k\rangle$ nach der idealen Anzahl von k Schritten (wieder für $n = 3$ und $M = 2$). Was ist die ideale Anzahl von Schritten? (3 Punkte)
4. Das Gleiche wie in Aufgabe (3) für $n = 3$ und $M = 1$. (2 Punkte)
5. Zeichne für $n = 7$ und $M = 1$ die Wahrscheinlichkeit, eine Lösung zu messen, gegen die Anzahl der Iterationen. (3 Punkte)

### A. Hilfestellung

Die Simulation kann natürlich in einer beliebigen Sprache so wie Mathematica, Maple, Matlab, C etc. durchgeführt werden. Hier noch ein paar Tips zum Verwenden von Mathematica:

1. Mit der Hilfefunktion F1 kann man jederzeit auf das komplette Mathematica-Handbuch zurückgreifen. Die Befehle sind alle so mit Beispielen dokumentiert, dass man sich schnell einlernt.
2. Listen, Vektoren und Matrizen kann man gut mit der Funktion `Table` erzeugen, so ergibt z.B. `A=Table[ If[m+n==2 n,1,0],{m,1,5},{n,1,5}]` eine 5x5 Einheitsmatrix, die sich schön als Matrix mit dem Befehl `A//MatrixForm` darstellen läßt.
3. Für die Orakelfunktion kann es praktisch sein, den Befehl `MemberQ[Liste, value]` zu benutzen, der überprüft, ob `value` in der `Liste` enthalten ist.
4. Datenpunkte (wie für Aufgabe (5)) benötigt, lassen sich z.B. mit `ListPlot[Liste]` darstellen.
5. Die Definition von Funktionen ist hier an einem Beispiel gezeigt: Wenn man
   `MyUnit[mfinal_,nfinal_,diagonal_]` :=Table[ If[m+n==2n,diagonal,0] ,{m,1,mfinal},{n,1,nfinal}]
   definiert hat, erzeugt `A=MyUnit[4,4,3]` eine 4x4 Diagonal-Matrix mit Diagonaleinträgen 3.
6. Das Produkt zwischen Matrizen und/oder Vektoren ist mit einem schlichten Punkt "." gekennzeichnet.

*bei Fragen: henning.moritz@physik.uni-hamburg.de