

Aufgabe 2.1 (4 Punkte)

Implementieren Sie einen Algorithmus, der in konstanter Zeit eine Rechtsrotation $\langle a, \dots, b, c \rangle \rightarrow \langle c, a, \dots, b \rangle$ einer Folge um eine Position durchführt. Die Folge ist in einer doppelt verketteten Liste gespeichert.

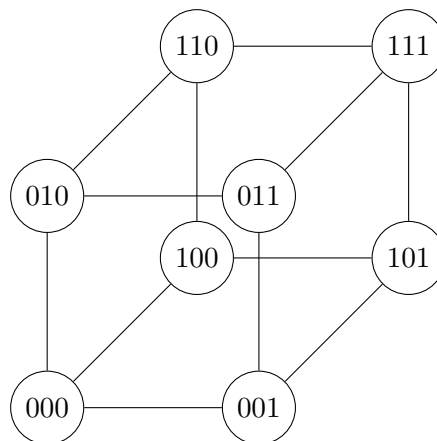
Verallgemeinern Sie den Algorithmus auf den Fall einer Rotation $\langle a, \dots, b, c, \dots, d \rangle \rightarrow \langle c, \dots, d, a, \dots, b \rangle$. Die Position von b ist dabei als Griff gegeben; die Ausführungszeit soll wieder konstant sein.

Aufgabe 2.2 (4 Punkte)

Erklären Sie, wie man eine Warteschlange mit Hilfe zweier Stapel implementieren kann, so dass jede Warteschlangen-Operation (PUSHBACK, POPFRONT) konstanten amortisierten Zeitbedarf hat.

Aufgabe 2.3 (4 Punkte)

Betrachten Sie den Graphen Q_3 (den 3-dimensionalen Würfel)



- Geben Sie die Knoten- und Kantenmenge an.
- Geben Sie die Nachbarschaft von Knoten 001 an, also alle zu 001 adjazenten Knoten. Wie lauten die Grade der Knoten?
- Geben Sie einen Kreis an, der den Knoten 111 enthält. Geben Sie einen Pfad von 011 nach 100 an.

Aufgabe 2.4 (4 Punkte)

Finden Sie für beliebiges n einen DAG mit n Knoten, der $n(n-1)/2$ Kanten besitzt. Zeigen Sie, dass kein DAG mit n Knoten mehr als $n(n-1)/2$ Kanten haben kann.