Übungsblatt 4: Sommersemester 2016

Programmiertechniken

Prof. Dr. L. Pollet J. Greitemann, D. Hügel, J. Nespolo, T. Pfeffer

---

1) Skalierung von typischen Behälteroperationen der Standardbibliothek

In dieser Aufgabe soll die Skalierung der Behälterklassen `std::vector` und `std::list` bezüglich verschiedener Operationen untersucht werden.

(a) Initialisieren Sie einen `std::vector` der Länge L mit Einträgen $v_i = 2i + 1$. Kopieren Sie den Inhalt des Vektors in einen Behälter der Klasse `std::list`.

(b) Nutzen Sie die Standardbibliothek, um die Laufzeit von verschiedenen Programmteilen zu messen.

(c) Untersuchen Sie die Skalierung der Rechenzeit bzgl. der Läenge L des Vektors für des Einfügens und das konsekutiven Löschens eines Eintrags in der Mitte der Behälters. Um einen guten Mittelwert zu bekommen, verwenden Sie mindestens 10^5 Operationen.

(d) Untersuchen Sie die Skalierung der Rechenzeit bzgl. der Läenge L des Vektors für das Auslesens eines beliebig gezogenen Elements aus den beiden Behältern. Die Zahl der Operation sollte wiederum 10^5 nicht unterschreiten.
Tipp: Die Standardbibliothek implementiert verschiedenste Zufallsgeneratoren.

Visualisieren Sie die Ergebnisse erneut mit Python. Stimmen die Ergebnisse mit der Theorie der Vorlesung überein?

2) Hashtables der Standardbibliothek

In der Vorlesung wurde besprochen wie die darstellende Matrix eines linearen Operators auf einem endlichdimensionalen Hilbertraum mit Hilfe einer Hashtable konstruiert werden kann. Als Beispiel wurde eine lineare Kette von Spins der Länge L betrachtet. Der Hilbertraum dieses Systems ist das Produkt der lokalen Hilberträume der einzelnen Spins wobei spin-down als $|0>$ und spin-up als $|1>$ dargestellt wird. Ein beliebiger Basiszustand des Gesamthilbertraums kann als Bitfolge $|101\ldots11>$ der Länge L dargestellt werden. Ist der Gesamtspin erhalten und auf m=0 fixiert so muss die Quersumme der Bitfolge gleich $\frac{L}{2}$ sein. Im folgendem wird L=8 fixiert.

1. Finden Sie alle Basisvektoren mit m=0 indem Sie alle möglichen Basisvektoren erzeugen und auf deren Quersumme überprüfen. Wie viele Basisvektoren gibt es mit dieser Bedingung und stimmt dies mit ihrer Erwartung überein?

2. Zu jedem Basisvektor gibt es einen eindeutigen Schlüssel definiert als die Dezimaldarstellung der Bitfolge. Finden Sie zu jedem Basisvektor den dazugehörigen Schlüssel.

Nicht jeder lineare Operator, dessen darstellende Matrix bzgl. einer gegebenen Basis gesucht wir, ist diagonal in dieser Basis. Zum Beispiel, siehe Folien der Vorlesung, verknüpft das Produkt von Aufsteige- S_i^+ und Absteigeoperator S_{i+1}^- den Zustand $|0011>$(Schlüssel 3, Index 0) mit dem Zustand $|0101>$ (Schlüssel 5, Index 1). Damit hat die darstellende Matrix dieses Operators an der Stelle (1,0) den Eintrag 1. Um die darstellende Matrix eines beliebigen Operators effektiv zu konstruieren muss für einen belieben Basisvektor der Index gefunden werden, welcher zu diesem Basisvektor korrespondiert.
Erzeugen Sie einen beliebigen Basisvektor.

1. Finden Sie den Index des Basisvektors durch eine lineare Suche in der Liste der möglichen Basisvektoren. Diese Methode skaliert O(N).

2. Finden Sie den Index des Basisvektors durch eine binäre Suche in einer nach den Schlüsseln geordneten Liste der Basisvektoren. Diese Methode skaliert O(log(N)).
   Tipp: Verwenden Sie das sortieren nach den Argumenten aus dem letzten Übungsblatt.

3. Finden Sie den Index des Basisvektors durch das Auslesen des Inhalts aus einer `std::unordered_map`. Diese Methode skaliert O(1).

Um die Funktionsweise und die Skalierung der `std::unordered_map` zu verstehen implementieren Sie diese Datenstruktur selbst.
Tipp: Versuchen Sie zuerst das Beispiel aus der Vorlesung zu verstehen und zu implementieren.