

Realisieren Sie eine Klasse *Dictionary* zum Speichern von Wortpaaren (deutsch,englisch) und zum Nachschlagen von englischen Übersetzungen deutscher Wörter. Die Wortpaare sollen in einer Hash-Tabelle mit offener Adressierung (offenes Hashverfahren) abgespeichert werden, wobei Kollisionen durch quadratisches Sondieren aufgelöst werden sollen.

Folgende Methoden werden verlangt:

- Konstruktor(n) legt eine Hashtabelle der Größe n an
- Destruktor
- insert(const string& dt, const string& engl) fügt das Wortpaar (dt,engl) in die Hashtabelle ein. Liefert 0, falls dt bereits vorhanden, 1 sonst.
- lookup(const string& dt, string& engl) schlägt zu dem deutschen Begriff dt den englischen Begriff engl nach. Liefert 1, falls dt gefunden, 0 sonst.
- del(const string& dt) löscht Eintrag mit dem deutschen Begriff dt. Liefert 1, falls (dt,engl) gelöscht werden konnte, 0 sonst.

- a) Schreiben Sie ein Hauptprogramm, das von einer Datei eine Folge von Wortpaaren (deutsch,englisch) einliest und in die Hashtabelle einfügt. Eine Beispieldatei mit etwa 16000 Einträgen finden Sie unter meiner Homepage. Wählen Sie die Hashtabelle etwa doppelt so groß. Beachten Sie, daß eine prime Hashtabellengröße (d.h. Hashtabellengröße ist eine Primzahl) das Hashverfahren erheblich verbessern kann.

Hinweis: Sollten Sie Probleme mit der Speicherplatzreservierung haben, verwenden Sie nur einen Teil der Beispieldatei.

- b) Erweitern Sie Ihre Klasse um eine Methode

double probingLength(int s);

die folgendes berechnen soll:

- 1)  $s = 1$ : die durchschnittliche Länge der Sondierungsfolgen bei erfolgreicher Suche.

Hinweis: Ermitteln Sie für jeden Eintrag der Hashtabelle die Länge der Sondierungsfolge für einen lookup-Aufruf und berechnen Sie daraus den durchschnittlichen Wert.

- 2)  $s = 0$ : die durchschnittliche Länge der Sondierungsfolgen bei nicht erfolgreicher Suche.

Hinweis: Ermitteln Sie für jedes englische Wort der Hashtabelle die Länge der Sondierungsfolge für einen lookup-Aufruf (sehr wahrscheinlich liefern fast alle lookup-Aufrufe „nicht gefunden“ zurück). Berechnen Sie daraus den durchschnittlichen Wert.

- c) Vergleichen Sie Ihre gemessenen Werte mit den theoretisch erreichbaren Werten (siehe Vorlesung). Beurteilen Sie damit Ihre Hashfunktion.

### Abgabe:

- Vorführen und Erklären des Programms
- Gemessene Werte aus b)
- Vergleich mit den theoretisch erreichbaren Werten