

Einführung in die Programmierung für Studierende der Naturwissenschaften

Blatt 7 – 20.06.2022

Abgabe: bis Montag, 27. Juni, 16 Uhr per e-Mail an Ihre:n Tutor:in.

Aufgabe 1 (5+5 Punkte). Gegeben Sei eine sortierte Liste von N Zahlen, d.h. eine Liste von Zahlen

$$e_0, \dots, e_{N-1}$$

s.d. $e_n < e_{n+1}$. Ein Verfahren, mit dem mit logarithmischem Aufwand getestet werden kann ob ein Element e in der Liste enthalten ist, funktioniert durch wiederholtes Halbieren der Liste und Suche nur in der passenden Teil-Liste.

- Schreiben Sie das Verfahren als pseudo-code auf und zeigen Sie, dass der Aufwand logarithmisch ist, also $\mathcal{A}(N) = C \cdot \log_2(N)$. Dabei dürfen Sie annehmen, dass N eine Zweierpotenz ist, d.h. $N = 2^n$ für ein $n \in \mathbb{N}$. Konstanten, die nicht von N abhängen, können Sie dabei ignorieren.
- Implementieren Sie das Verfahren für sortierte Listen beliebiger Länge N , also so dass N nicht notwendigerweise eine Zweierpotenz sein muss.

Aufgabe 2 (4 + 6 Punkte). Auf der Vorlesungswebseite finden Sie eine Datei `merge_sort_vector.cc`, in der das Mergesort-Verfahren so implementiert wurde, dass damit `std::vector<int>` beliebiger Länge sortiert werden können.

- Ändern Sie die Implementierung so ab, dass diese ein *Template* wird. Damit sollen `std::vector<T>` beliebigen Typs T sortiert werden können, für die ein `<=` Vergleichs-Operator definiert ist.
- Ändern Sie die Implementierung erneut, so dass die Methoden einen Funktionspointer `bool(*compare)(T, T)` als zusätzliches Argument erhalten. Anstatt mit dem `<=` Operator sollen je zwei Elemente dann mit der `compare` Funktion verglichen werden. In der sortierten Liste soll ein Element `e1` vor einem Element `e2` stehen, falls `compare(e1, e2) == true` ist. Testen Sie Ihre Implementierung, in dem Sie die Elemente eines `std::vector<std::string>` sortieren und mit einer Methode

```
1 bool wort_ist_kuerzer(std::string w1, std::string w2) {
2     return w1.length() < w2.length();
3 }
```

vergleichen. Die Wörter in dem `std::vector<string>` sollten damit nach der Länge aufsteigend sortiert werden.

Hinweis: Wenn Sie beide Aufgabenteile lösen reicht es, wenn Sie Ihre Lösung zu (b) die (a) enthält abgeben. Trotz sorgfältiger Implementierung und Überprüfung können im vorgegebenen Quelltext natürlich Fehler enthalten sein. Falls Sie einen finden und uns mitteilen, so dass

wir das Problem beheben können, bekommen Sie 5 Bonus Punkte. Beachten Sie, dass die Implementierung nicht exakt der aus der Vorlesung entspricht. Dies ist für die Aufgabenstellung aber nicht relevant.

Aufgabe 3 (optional). Schildern Sie in einer Datei `erfahrung.txt` kurz Ihre Erfahrung mit dem aktuellen Übungsblatt. Berichten Sie darin wieder in Stichpunkten bzw. ein bis zwei kurzen Sätzen über Ihre Erfahrungen mit Kursinhalt und Übungsaufgaben. Was fiel Ihnen leicht? Was ist noch unklar? Wie viel Zeit haben Sie für die Bearbeitung der Hausaufgaben benötigt und welche Probleme traten dabei auf?