



KLAUSUR ZUR PROGRAMMIERUNG II, SS 2003

Hinweise zur Klausur

- Sie haben zum Bearbeiten der Aufgaben zwei Stunden Zeit. Die Punktzahlen der einzelnen Aufgaben entsprechen der Zeit (in Minuten), die Sie für die Bearbeitung der Aufgabe maximal aufwenden sollten, wenn Sie alle Aufgaben bearbeiten wollen.
- Es können maximal 120 Punkte erreicht werden. Ab 60 Punkten gilt die Klausur als bestanden.
- Bitte schreiben Sie auf dieses Deckblatt Ihren Namen, Vornamen, Geburtsdatum und Matrikelnummer. Unterschreiben Sie diese Angaben und geben Sie dieses Blatt mit Ihren Lösungen ab. Versehen Sie jeden abgegebenen Papierbogen mit Ihrem Namen und geben Sie bitte auch Ihr Schmierpapier mit ab.
- Legen Sie Ihren Personalausweis und Studierendenausweis zur Kontrolle auf den Tisch.
- Kreuzen Sie in der Tabelle auf diesem Blatt die von Ihnen bearbeiteten Aufgaben an.
- **Überprüfen Sie, ob Sie einen vollständigen Klausurbogen erhalten haben. Sie müssten 13 Blätter mit 8 Aufgaben bekommen haben.**

Name:

Vorname:

Geburtsdatum:

Matrikelnummer:

Unterschrift:

Aufgabe	1	2	3	4	5	6	7	8	Σ
bearbeitet									
Punkte									

Viel Erfolg!

Aufgabe 1: Wissenstest (Insgesamt 21 Punkte)

Hinweis: Bei dieser Aufgabe müssen Sie Ihre Antworten ausnahmsweise **nicht** begründen!

- (a) Wann nennt man eine C++-Klasse ein *Funktionsobjekt*? (3 Punkte)
 - (b) Definieren Sie $\omega(n)$ (3 Punkte)
 - (c) Was versteht man beim Hashing unter *Chaining*? (3 Punkte)
 - (d) Welche Eigenschaften definieren einen *Rot-Schwarz-Baum*? (4 Punkte)
 - (e) Was versteht man unter einem *ungerichteten Euler-Graphen*? (4 Punkte)
 - (f) Beschreiben Sie kurz den Algorithmus zum topologischen Sortieren. (4 Punkte)
-

Aufgabe 2: C++ (20 Punkte)

Erstellen Sie eine C++-Templateklasse `liste<T>`, die eine doppelt verkettete Liste implementiert. Geben Sie dazu die Klassendeklaration möglichst vollständig an (inklusive Konstruktor, Copy-Konstruktor, Destruktor und Assignment-Operator). Ihre Klasse soll ausserdem die Operationen *Löschen eines Elements*, *Einfügen eines Elements* und *Suchen nach einem Element* bereitstellen und eine Hilfsklasse (als *nested class*) enthalten, die zur Repräsentation der Listenelemente verwendet werden kann. Geben Sie zusätzlich die Implementierung der Methode `bool delete(const T& i)` an, welche das Element i aus der Liste löscht und `true` zurückliefert, falls i in der Liste enthalten war. War das Element i nicht enthalten, liefert die Funktion `false` zurück. Sie dürfen dabei davon ausgehen, dass kein Element mehr als einmal in der Liste vorkommt!

Aufgabe 3: $\mathcal{O}$ -Notation (10 Punkte)

Finden Sie eine Anordnung der folgenden Funktionen, so dass gilt $f_i \in \mathcal{O}(f_{i+1}) \forall i$ und geben Sie für jedes Paar f_i, f_{i+1} eine kurze Begründung Ihrer Anordnung an.

(a) $n!$

(b) e^n

(c) $4^{\log_2(n+1)}$

(d) $n \ln(n)$

Aufgabe 4: Substitutionsmethode (11 Punkte)

Lösen Sie die folgende Rekurrenzgleichung mit Hilfe der Substitutionsmethode, d.h., geben Sie ein $f(n)$ an mit: $T(n) \in \Theta(f(n))$. Beweisen Sie: $T(n) \in \mathcal{O}(f(n))$. Das auch gilt $T(n) \in \Omega(f(n))$, müssen Sie hier ausnahmsweise *nicht* beweisen!

$$T(n) = 2T\left(\frac{n}{3}\right) + n^3$$

Aufgabe 5: Mastertheorem (10 Punkte)

Zeigen Sie für jede der folgenden Rekurrenzen ob das Mastertheorem zur Lösung angewandt werden kann. Bestimmen Sie in den Fällen, in denen sich das Theorem anwenden lässt, die Lösung der Rekurrenz.

- $T(n) = 64T(\frac{n}{8}) + 1042n^2 + 1000000n$
- $T(n) = 27T(\frac{n}{3}) + n^3 \ln(n) + 42$

Aufgabe 6: Erwartungswert (13 Punkte)

Betrachten Sie ein Gewinnspiel, das beliebig oft wiederholt werden darf. Dabei beschreibt die Zufallsvariable X die Anzahl an Versuchen, die Sie durchführen müssen, bevor Sie den Gewinn kassieren können, d.h., $X = n$ bedeutet, dass die ersten $n - 1$ Versuche erfolglos, der n -te hingegen erfolgreich war. Die Wahrscheinlichkeit $p(\{X \geq i\})$ dafür, dass Sie mindestens i Versuche benötigen, ist Ihnen bekannt und es gilt:

$$p(\{X \geq i\}) = \left(\frac{1}{\alpha}\right)^i, \alpha > 1$$

Zeigen Sie: $E[X] = \frac{1}{\alpha-1}$.

Aufgabe 7: Bäume (17 Punkte)

Zwei Mengen A und B seien in den zugehörigen 2–4 - Bäumen T_A und T_B gespeichert. Beschreiben Sie einen Algorithmus, der den 2–4 - Baum $T_{A \cup B}$ der vereinigten Menge $A \cup B$ in Zeit $\mathcal{O}(|A| + |B|)$ berechnet. (Es reicht aus, wenn Sie Ihren Algorithmus in Pseudocode formulieren!) Begründen Sie Korrektheit und Laufzeit Ihres Algorithmus!

Hinweis: Zeigen Sie dazu zuerst, dass Sie den 2–4 - Baum einer schon sortierten Sequenz $a_1, \dots, a_{n-1}$ in Zeit $\mathcal{O}(n)$ aufbauen können.

Aufgabe 8: Graphen (18 Punkte)

Zeigen Sie, dass jeder ungerichtete zusammenhängende Graph $G = (V, E)$ mindestens einen Knoten enthält, bei dessen Entfernung aus G der Graph zusammenhängend bleibt (wenn Sie den Knoten entfernen, entfernen Sie natürlich auch die zu ihm gehörigen Kanten aus dem Graphen). Entwickeln Sie einen Algorithmus, der einen solchen Knoten bestimmt und eine worst-case Laufzeit von $\mathcal{O}(|V| + |E|)$ aufweist (es reicht aus, wenn Sie diesen in Pseudocode formulieren!). Beweisen Sie die Korrektheit Ihres Verfahrens und begründen Sie kurz dessen Laufzeit.

Für Ihre Zwischenrechnungen

Für Ihre Zwischenrechnungen

Für Ihre Zwischenrechnungen