Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Institut für Theoretische Informatik
Peter Widmayer
Tobias Pröger

Prüfung
Datenstrukturen und Algorithmen
D-INFK

25. Januar 2014

Name, Vorname: ______________________________

Stud.-Nummer: ______________________________

Ich bestätige mit meiner Unterschrift, dass ich diese Prüfung unter regulären Bedingungen ablegen konnte und dass ich die untenstehenden Hinweise gelesen und verstanden habe.

Unterschrift: ______________________________

Hinweise:

- Ausser einem Fremdsprachenwörterbuch dürfen Sie keine Hilfsmittel verwenden.
- Bitte schreiben Sie Ihre Studierenden-Nummer auf **jedes** Blatt.
- Melden Sie sich bitte **sofort**, wenn Sie sich während der Prüfung in irgendeiner Weise bei der Arbeit gestört fühlen.
- Bitte verwenden Sie für jede Aufgabe ein neues Blatt. Pro Aufgabe kann nur eine Lösung angegeben werden. Ungültige Lösungsversuche müssen klar durchgestrichen werden.
- Bitte schreiben Sie **lesbar** mit blauer oder schwarzer Tinte. Wir werden nur bewerten, was wir lesen können.
- Sie dürfen alle Algorithmen und Datenstrukturen aus der Vorlesung verwenden, ohne sie noch einmal zu beschreiben. Wenn Sie sie modifizieren, reicht es, die Modifikationen zu beschreiben.
- Die Prüfung dauert 180 Minuten.

Viel Erfolg!

Stud.-Nummer: ______________________________

Aufgabe	1	2	3	4	5	$\boldsymbol{\Sigma}$
Mögl. Punkte	9	8	8	12	13	50
$\sum$ Punkte						

Aufgabe 1.

Hinweise:

1) In dieser Aufgabe sollen Sie **nur die Ergebnisse** angeben. Diese können Sie direkt bei den Aufgaben notieren.
2) Sofern Sie die Notationen, Algorithmen und Datenstrukturen aus der Vorlesung "Datenstrukturen & Algorithmen" verwenden, sind Erklärungen oder Begründungen nicht notwendig. Falls Sie jedoch andere Methoden benutzen, müssen Sie diese **kurz** soweit erklären, dass Ihre Ergebnisse verständlich und nachvollziehbar sind.
3) Als Ordnung verwenden wir für Buchstaben die alphabetische Reihenfolge, für Zahlen die aufsteigende Anordnung gemäss ihrer Grösse.

1 P (a) Im untenstehenden Array sind die Elemente eines Max-Heaps in der üblichen Form gespeichert. Wie sieht das Array aus, nachdem das Maximum entfernt wurde und die Heap-Bedingung wieder hergestellt wurde?

27	17	20	15	7	9	13	8	2	5	3	1	6
1	2	3	4	5	6	7	8	9	10	11	12	13

1 P (b) Gegeben sei ein zusammenhängender, ungerichteter Graph $G = (V, E)$ mit $n = |V|$ Knoten und $m = |E|$ Kanten. Wie viele Knoten und Kanten besitzt ein Spannbaum von G?

Knoten: ________ Kanten: ________

1 P (c) Zeichnen Sie einen zusammenhängenden Graphen mit 7 Knoten, der ein maximales Matching der Grösse 2 besitzt.

3 P (d) Kreuzen Sie an, ob die folgenden Aussagen wahr oder falsch sind. Jede korrekte Antwort gibt 0,5 Punkte, für jede falsche Antwort werden 0,5 Punkte abgezogen. Eine fehlende Antwort gibt 0 Punkte. Insgesamt gibt die Aufgabe mindestens 0 Punkte. Sie müssen Ihre Antworten nicht begründen.

Eine Postorder-Traversierung eines binären Suchbaums erzeugt eine absteigend sortierte Liste der gespeicherten Schlüssel.	☐ Wahr	☐ Falsch
Es gibt einen AVL-Baum, bei dem mehr als die Hälfte seiner inneren Knoten nicht perfekt balanciert sind (d.h., einen Balancierungsfaktor ungleich 0 haben).	☐ Wahr	☐ Falsch
Zu jedem AVL-Baum gibt es eine Einfügereihenfolge, die zu genau diesem Baum führt, ohne dass Rotationen stattfinden.	☐ Wahr	☐ Falsch
Das Einfügen eines neuen Schlüssels in einen AVL-Baum führt selbst im schlimmsten Fall zu nur einer (einfachen oder Doppel-)Rotation.	☐ Wahr	☐ Falsch
Das Einfügen eines neuen Elements in einen Fibonacci-Heap erfordert im schlimmsten Fall nur konstante Zeit.	☐ Wahr	☐ Falsch
Sortieren durch Einfügen kann als stabiles Sortierverfahren implementiert werden.	☐ Wahr	☐ Falsch

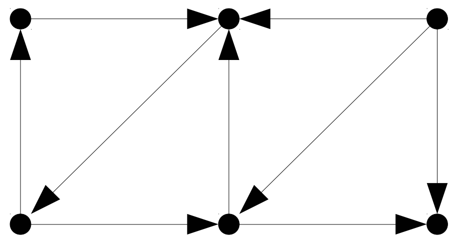
1 P (e) Gegeben sei ein Segmentbaum zur Verwaltung ganzzahliger Intervalle mit Intervallgrenzen aus $\{1,\ldots,n+1\}$ für eine Zweierpotenz $n = 2^k$, $k \in \mathbb{N}$. Geben Sie die Anzahl Knoten an, die eine Aufspiessanfrage für einen Punkt $i \in \{1,\ldots,n+1\}$ maximal besucht.

Maximale Anzahl: ______

1 P (f) Fügen Sie die Schlüssel 12, 19, 6, 15, 13, 2, 28 in dieser Reihenfolge in die untenstehende Hashtabelle ein. Benutzen Sie Double Hashing mit der Hashfunktion $h(k) = k \bmod 11$, und benutzen Sie $h'(k) = 1 + (k \bmod 9)$ zur Sondierung.

0	1	2	3	4	5	6	7	8	9	10

1 P (g) Markieren Sie in untenstehendem Graphen $G = (V, E)$ eine *kleinstmögliche* Menge S an Kanten, sodass der Graph $G' = (V, E \backslash S)$ topologisch sortiert werden kann.



Aufgabe 2.

1 P (a) Geben Sie für die untenstehenden Funktionen eine **Reihenfolge** an, so dass folgendes gilt: Wenn eine Funktion f links von einer Funktion g steht, dann gilt $f \in \mathcal{O}(g)$.

Beispiel: Die drei Funktionen n^3, n^7, n^9 sind bereits in der entsprechenden Reihenfolge, da $n^3 \in \mathcal{O}(n^7)$ und $n^7 \in \mathcal{O}(n^9)$ gilt.

- $\frac{3^n}{n^3}$
- 10^{10}
- $\log(n^n)$
- $n!$
- $n^3 + n$
- $\sqrt{3^n}$

4 P (b) Gegeben ist die folgende Rekursionsgleichung:

$$T(n) := \begin{cases} 5T(n/5) + n + 4 & n > 1 \\ 1 & n = 1 \end{cases}$$

Geben Sie eine geschlossene (d.h. nicht-rekursive) und *möglichst einfache* Formel für $T(n)$ an und beweisen Sie diese mit vollständiger Induktion.

Hinweise:

(1) Sie können annehmen, dass n eine Potenz von 5 ist.

(2) Für $q \neq 1$ gilt: $\sum_{i=0}^{k} q^i = \frac{q^{k+1}-1}{q-1}$.

1 P (c) Geben Sie die asymptotische Laufzeit in Abhängigkeit von $n \in \mathbb{N}$ für das folgende Programmstück (so knapp wie möglich) in Θ-Notation an. Sie müssen Ihre Antwort nicht begründen.

```
for(int i = 1; i <= n; i += 10) {
        for(int j = 1; j <= n/2; j += 4)
                ;
}
```

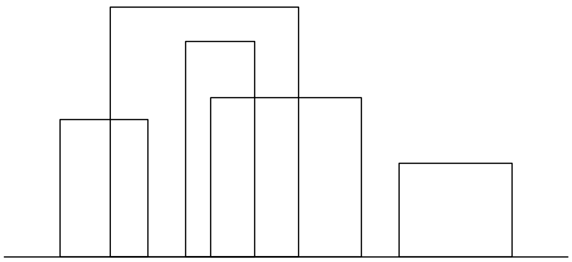
1 P (d) Geben Sie die asymptotische Laufzeit in Abhängigkeit von $n \in \mathbb{N}$ für das folgende Programmstück (so knapp wie möglich) in Θ-Notation an. Sie müssen Ihre Antwort nicht begründen.

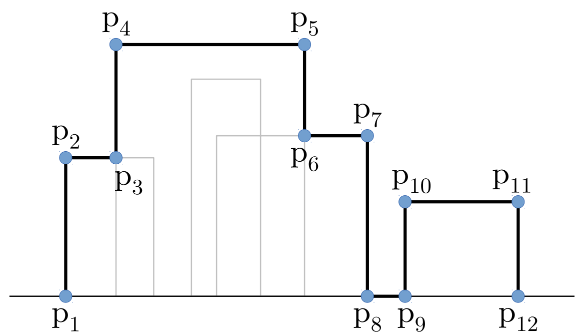
```
for(int i = 1; i <= n; i++) {
        for(int j = 1; j*j <= n; j++)
                ;
        for(int k = n; k ≥ 2; k /= 2)
                ;
}
```

1 P (e) Geben Sie die asymptotische Laufzeit in Abhängigkeit von $n \in \mathbb{N}$ für das folgende Programmstück (so knapp wie möglich) in Θ-Notation an. Sie müssen Ihre Antwort nicht begründen, und dürfen weiterhin der Einfachheit halber annehmen, dass n eine Potenz von 2 ist.

```
int f(int n) {
        if(n <= 1) { return 1; }
        else { return f(n/2)+f(n/2); }
}
```

Aufgabe 3. In dieser Aufgabe geht es um die Berechnung des Umrisses von Skylines. Die Skyline einer Stadt (der Umriss) ergibt sich wie der Schatten einer Menge rechteckiger Hochhäuser, die allesamt auf dem Boden aufsitzen und als orthogonale Recktecke wie im Bild links gegeben sind. Der Umriss ist das orthogonale Polygon, das die Vereinigung aller solchen gegebenen Rechtecke beschreibt, wie im Bild rechts zu sehen. Er kann durch die Menge seiner Eckpunkte $p_1, \ldots, p_k$ beschrieben werden.





7 P a) Entwerfen Sie einen möglichst effizienten Scanline-Algorithmus, der als Eingabe eine Menge von n orthogonalen Rechtecken erhält, und die Eckpunkte $p_1, \ldots, p_k$ des Umrisses berechnet. Gehen Sie in Ihrer Lösung auf die folgenden Aspekte ein.

1) In welche Richtung verläuft die Scanline, und was sind die Haltepunkte?
2) Welche Objekte muss die Datenstruktur verwalten, und was ist eine angemessene Datenstruktur?
3) Was passiert, wenn die Scanline auf einen neuen Haltepunkt trifft?
4) Wie lässt sich die Lösung auslesen?

1 P b) Geben Sie die Laufzeit Ihres in a) entwickelten Algorithmus an und begründen Sie Ihre Antwort.

Aufgabe 4. Ein Versandhandel bietet die Artikeltypen $\{1, \dots, n\}$ an. Wir bestellen von jedem Artikeltyp einige Exemplare, und beauftragen für den Transport $m \leq n$ Kuriere. Allerdings transportiert nicht jeder Kurier jeden Artikeltypen (z.B. kann ein Fahrradkurier keinen Fernseher ausliefern). Für jeden Kurier $j \in \{1, \dots, m\}$ sei $T(j) \subseteq \{1, \dots, n\}$ die Menge der Artikeltypen, die er transportieren kann. Wir möchten entscheiden, ob die Artikel(exemplare) so auf die Kuriere aufgeteilt werden können, dass alle Artikel befördert werden und jeder Kurier nur einmal fahren muss.

4 P a) Angenommen, von jedem Artikeltyp $i \in \{1, \dots, n\}$ wird genau ein Exemplar bestellt, und jeder Kurier $j \in \{1, \dots, m\}$ soll pro Fahrt nur ein einziges Artikelexemplar transportieren. Modellieren Sie das o.g. Problem als Flussproblem. Beschreiben Sie dazu die Konstruktion eines geeigneten Netzwerks $G = (V, E, c)$ mit der Knotenmenge V sowie der Kantenmenge E, und welche Kapazitäten den Kanten zugewiesen werden. Wie kann aus dem Wert eines maximalen Flusses abgelesen werden, ob eine geeignete Verteilung existiert oder nicht?

3 P b) Nun wird von jedem Artikeltyp $i \in \{1, \dots, n\}$ nicht nur ein, sondern $f(i) \in \mathbb{N}_0$ Exemplare bestellt, und jeder Kurier $j \in \{1, \dots, m\}$ kann pro Fahrt nicht nur einen, sondern bis zu $k(j)$ Artikelexemplare insgesamt befördern. Beschreiben Sie, wie Sie die Lösung aus a) modifizieren können, um dieses allgemeinere Problem zu lösen. Wie gross muss nun der Wert eines maximalen Flusses sein, damit jeder Kurier nur einmal fahren muss?

Beispiel:

i	Artikeltyp	Bestellte Exemplare $f(i)$
1	Tisch	2
2	Stuhl	2
3	Stift	5
4	Drucker	0

Kurier j	Max. Anz. Artikel $k(j)$ pro Fahrt	Akzeptierte Artikeltypen $T(l)$
1	4	$\{3\}$ (Stift)
2	3	$\{1, 2, 3, 4\}$ (Alles)
3	2	$\{1, 2\}$ (Tisch, Stuhl)

Hier reicht eine Fahrt pro Kurier, z.B. indem 4 Stifte von Kurier 1 transportiert werden, 1 Stift sowie 2 Tische von Kurier 2, und die 2 Stühle von Kurier 3.

2 P c) Nennen Sie einen Algorithmus, der das Problem aus b) möglichst effizient löst, und geben Sie die Laufzeit im schlimmsten Fall in Abhängigkeit von der Anzahl der Artikeltypen n und der Anzahl der Kuriere m an. Begründen Sie Ihre Antwort.

3 P d) Angenommen, das Flussproblem aus b) wurde bereits gelöst, d.h. zu einem maximalen Fluss ϕ kennen Sie den Fluss ϕ_e auf jeder Kante e, und angenommen, eine geeignete Verteilung wie oben beschrieben existiert tatsächlich. Beschreiben Sie detailliert einen Algorithmus, der aus den ϕ_e eine solche Verteilung berechnet. Konkret soll eine Menge M berechnet werden; diese enthält ein Tripel (j, i, k) genau dann, wenn der Kurier j genau k Exemplare vom Artikel i transportiert. Für das obige Beispiel ist $M = \{(1, 3, 4), (2, 1, 2), (2, 3, 1), (3, 2, 2)\}$. Welche Laufzeit hat Ihr Verfahren, wenn auf jedes ϕ_e in Zeit $\Theta(1)$ zugegriffen werden kann?

Aufgabe 5. In der Schweiz existieren die folgenden Münzwerte: 5 Rappen, 10 Rappen, 20 Rappen, 50 Rappen, 1 Franken, 2 Franken, 5 Franken. Damit lassen sich alle Geldbeträge darstellen, deren Wert in Rappen durch 5 teilbar ist. In dieser Aufgabe befassen wir uns mit der Anzahl verschiedener Möglichkeiten, einen gegebenen Betrag durch irgendeine Menge von Münzen zu erreichen. Beispielsweise kann der Betrag von 20 Rappen auf vier verschiedene Arten erreicht werden:

1) $5 + 5 + 5 + 5$ Rappen,
2) $5 + 5 + 10$ Rappen,
3) $10 + 10$ Rappen,
4) 20 Rappen.

Beachten Sie, dass es dabei auf die Reihenfolge der Münzen nicht ankommt; $5 + 5 + 10$ Rappen ist also dieselbe Menge von Münzen wie $5 + 10 + 5$ Rappen. Für diese Aufgabe gehen wir von der allgemeinen Annahme aus, dass n Münzen mit den Werten $M_1, \ldots, M_n \in \mathbb{N}$ (in der gleichen Einheit, z.B. Rappen) gegeben sind. O.B.d.A. seien diese aufsteigend geordnet und paarweise verschieden, d.h. $M_1 < M_2 < \cdots < M_n$. Für den Schweizer Franken z.B. hätten wir die Münzwerte $M_1 = 5$, $M_2 = 10$, $M_3 = 20$, $M_4 = 50$, $M_5 = 100$, $M_6 = 200$, $M_7 = 500$ (jeweils in Rappen ausgedrückt).

Hinweis: Die Münzwerte M_i sind nicht notwendigerweise Vielfache von 5 wie im obigen Beispiel, sondern beliebige Zahlen. Es könnten sogar alle M_i Primzahlen (d.h., paarweise teilerfremd) sein.

8 P a) Gegeben sei ein ganzzahliger Betrag $B \in \mathbb{N}$ in der gleichen Einheit wie die Münzen $M_1, \ldots, M_n$. Geben Sie einen möglichst effizienten Algorithmus an, der nach dem Prinzip der dynamischen Programmierung arbeitet und die Anzahl der Möglichkeiten, den Betrag B mit den Münzen $M_1, \ldots, M_n$ darzustellen, berechnet. Gehen Sie in Ihrer Lösung auf die folgenden Aspekte ein.

1) Was ist die Bedeutung eines Tabelleneintrags, und welche Grösse hat die DP-Tabelle?
2) Wie berechnet sich ein Tabelleneintrag aus früher berechneten Einträgen?
3) In welcher Reihenfolge müssen die Einträge berechnet werden?
4) Wie lässt sich die Lösung aus der DP-Tabelle extrahieren?

2 P b) Geben Sie die Laufzeit der Lösung aus Aufgabenteil a) an und begründen Sie Ihre Antwort. Ist die Laufzeit polynomiell?

3 P c) Beschreiben Sie detailliert, wie durch Rückverfolgung in der Lösungstabelle gemäss b) *eine* mögliche Darstellung des Betrags B gefunden werden kann, sofern überhaupt eine existiert. Geben Sie auch die Laufzeit dieses Algorithmus an.