UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Fakultät für Elektrotechnik, Informatik und Mathematik
Juniorprofessor Dr. Robert Elsässer

Klausur 1 - Datenstrukturen und Algorithmen SS 2009

Dienstag, 11. August 2009, 12-15Uhr

Name, Vorname: ....

Matrikelnummer: ....

Studiengang bitte ankreuzen:

☐ Informatik, Bachelor (seit WS 04/05)
☐ Informatik, Bachelor/Diplom (DPO 4)
☐ Informatik, Lehramt
☐ Ingenieurinformatik mit Schwerpunkt Informatik
☐ Ingenieurinformatik mit Schwerpunkt Elektrotechnik
☐ Ingenieurinformatik mit Schwerpunkt Maschinenbau
☐ Algorithmische Mathematik
☐ Mathematik Diplom
☐ Wirtschaftsinformatik
☐ Medienwissenschaften (80%)
☐ Magister mit Nebenfach Informatik (80%)
☐ Sonstiges: ....

Bitte Zutreffendes ankreuzen:

Fachprüfung: ☐ Leistungsnachweis: ☐ Teilnahmeschein: ☐

Bitte sorgfältig durchlesen und beachten!

- Prüfen Sie die Vollständigkeit Ihres Klausurexemplares (Deckblatt, 8 Aufgaben).
- Schreiben Sie zu Beginn der Klausur auf das Deckblatt und jede weitere Seite DEUTLICH LESBAR Ihren Namen und Ihre Matrikelnummer. KLAUSUREN OHNE DIESE ANGABEN WERDEN NICHT GEWERTET!
- Verwenden Sie keinen Rotstift, keinen Bleistift und kein eigenes Konzeptpapier. Benutzen Sie nur den Platz unter den Aufgabenstellungen und die Rückseiten der Blätter für Ihre Lösungen. Am Ende der Klausur befinden sich zwei Blätter Konzeptpapier, die Sie abreißen können.
- Sollten Sie zusätzliches Konzeptpapier benötigen, so wenden Sie sich bitte an die Aufsicht.
- Werden zu einer Aufgabe mehrere Lösungen angegeben, so gilt die Aufgabe als NICHT gelöst.
- Als Hilfsmittel ist NUR ein doppelseitig handbeschriebenes DIN A4 Blatt erlaubt; schreiben Sie auch auf dieses Blatt Ihren Namen.
- **Zulassungsvorbehalt:** Hätten Sie wissen können, dass Sie zur Klausur nicht zugelassen sind, wird die bestandene Klausur NICHT gewertet; die nicht bestandene gilt ggf. als Fehlversuch.

Aufgabe	1	2	3	4	5	6	7	8	$\sum$
erreichbare Punkte	7	6	8	6	6	6	8	7	54
erreichte Punkte									

Name:	Matrikelnummer:

AUFGABE 1: Multiple-Choice (7 Punkte)

Hinweis: Falsche Antworten führen zu Punktabzügen!

1. Die Matrixmultiplikation nach Strassen hat eine Laufzeit von:
   ☐ $\Theta(n^3)$ und nutzt 8 Multiplikationen pro Rekursionsschritt
   ☐ $\Theta(n^{2.87})$ und nutzt 7 Multiplikationen pro Rekursionsschritt
   ☐ $\Theta(n^{2.87})$ und nutzt 7 Additionen pro Rekursionsschritt
   ☐ $\Theta(n^{\log_2(7)})$ und nutzt 7 Multiplikationen pro Rekursionsschritt

2. Welche Strategie führt beim Intervall-Scheduling zu einer optimalen Lösung:
   ☐ wähle das Intervall mit der größten Ausführungslänge.
   ☐ wähle das noch verbliebene kürzeste Intervall, welches keine Inkompatibilität verursacht, hinzu.
   ☐ wähle das Intervall mit den wenigsten inkompatiblen Intervallen. Bei Gleichheit wähle das kürzeste Intervall.
   ☐ wähle das Intervall, welches am frühesten beendet ist.

3. Welcher der folgenden Algorithmen basiert auf dem Divide&Conquer-Prinzip?
   ☐ Insertion-Sort
   ☐ Counting-Sort
   ☐ Heap-Sort
   ☐ Quick-Sort

4. Im Worst-Case hat Merge-Sort Laufzeit von:
   ☐ $\Theta(n^2 \log n)$
   ☐ $\Theta(n^2)$
   ☐ $\Theta(n \log n)$
   ☐ $\Theta(n)$

5. Welcher der folgenden Algorithmen ist *kein* Greedy-Algorithmus?
   ☐ Prim
   ☐ Kruskal
   ☐ Floyd-Warshall
   ☐ Huffman

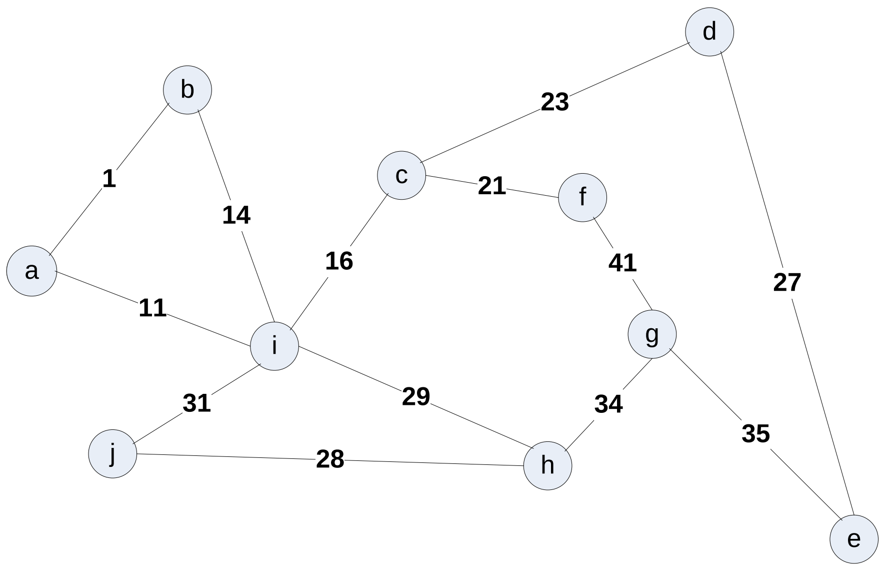
6. Für welchen der folgenden Texte ist a="0" b="10" c="11" eine optimale Präfixkodierung?
   ☐ a a b b c c
   ☐ a b b c c
   ☐ a a b b b c c
   ☐ a b c c

7. Welche Aussage ist nach dem Mastertheorem für die Rekursion $T(n) = 7 * T(\frac{n}{3}) + n^2$ korrekt?
   ☐ $T(n) = O(n^{\log_3 7})$
   ☐ $T(n) = O(n^{\log_3 7} * \log n)$
   ☐ $T(n) = \Theta(n^2)$
   ☐ Das Mastertheorem kann nicht angewendet werden.

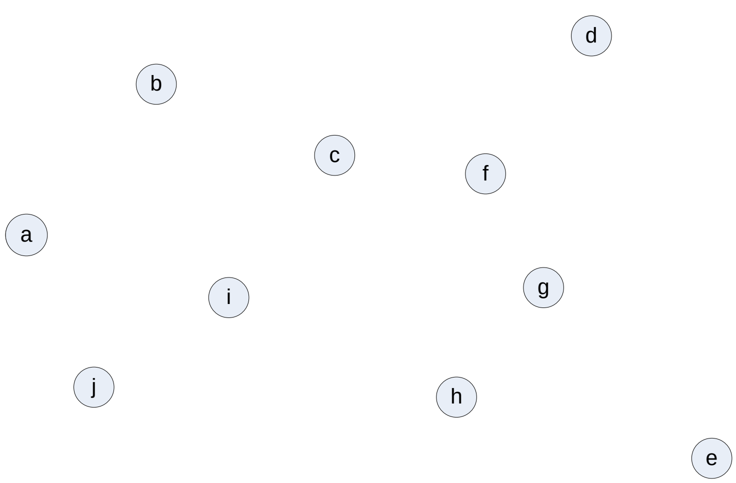
Name:	Matrikelnummer:

AUFGABE 2: Prim (6 Punkte)

Wenden Sie den Algorithmus von Prim auf den nachfolgenden Graphen mit Startknoten a an. Geben Sie hierzu nach jedem Schritt den jeweils vorliegenden Heap (in grafischer Darstellung), den Schnitt im Graphen und die jeweils für den MST verwendete Kante an.

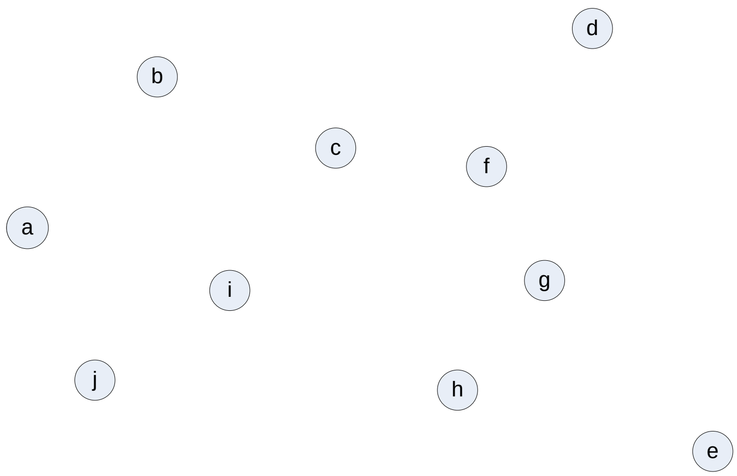


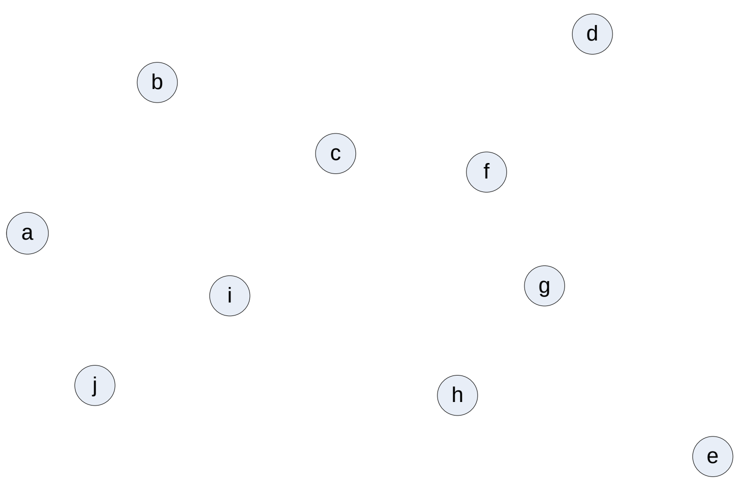
Graph:	Heap:

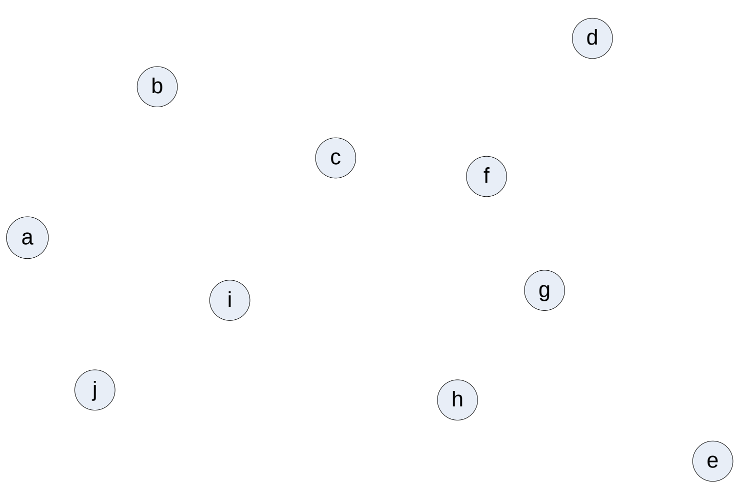


Name:	Matrikelnummer:

Graph:	Heap:

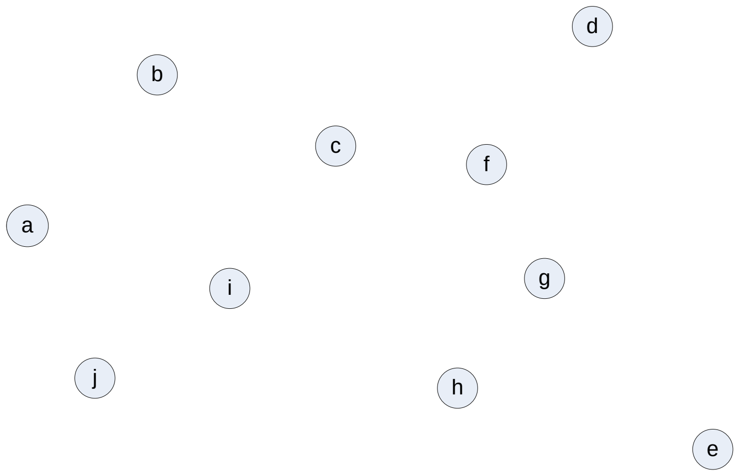


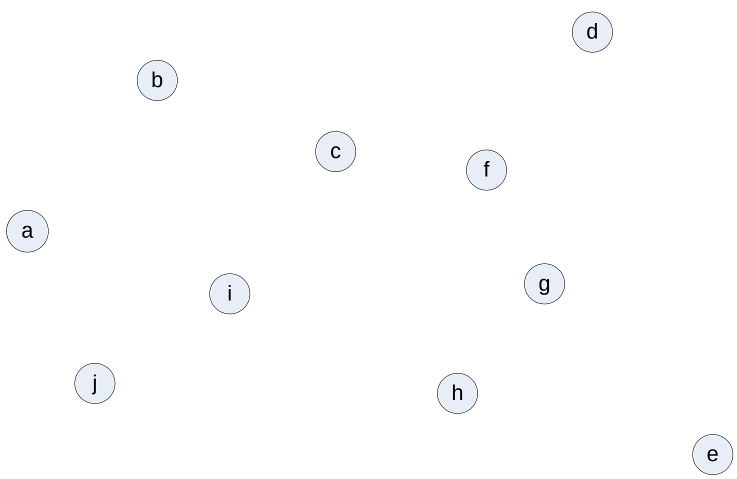


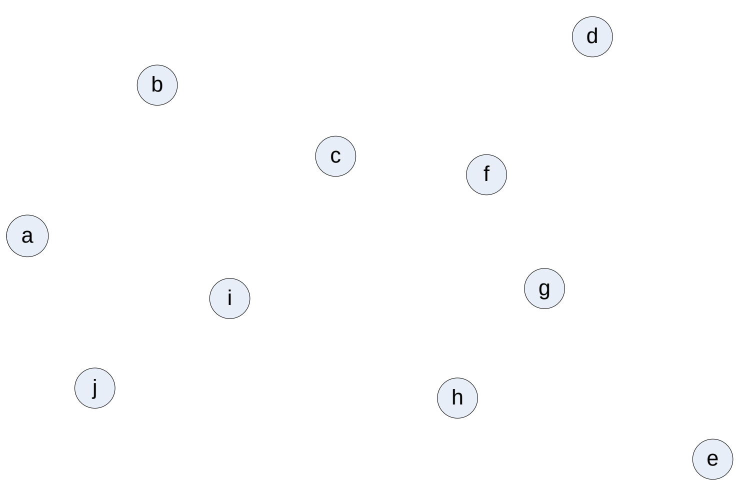


Name:	Matrikelnummer:

Graph:	Heap:

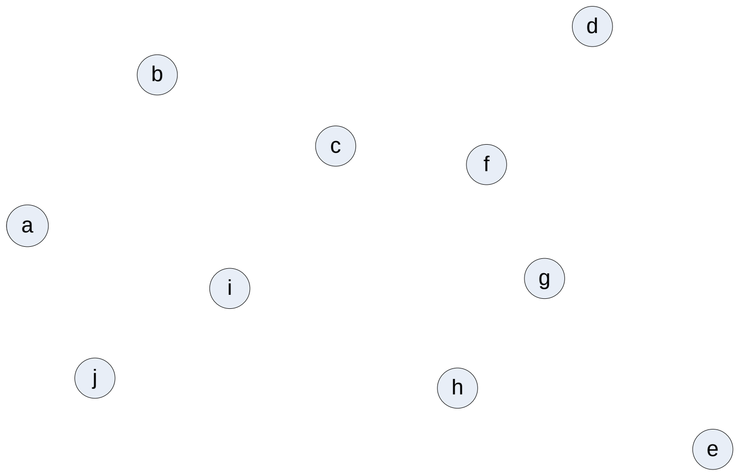


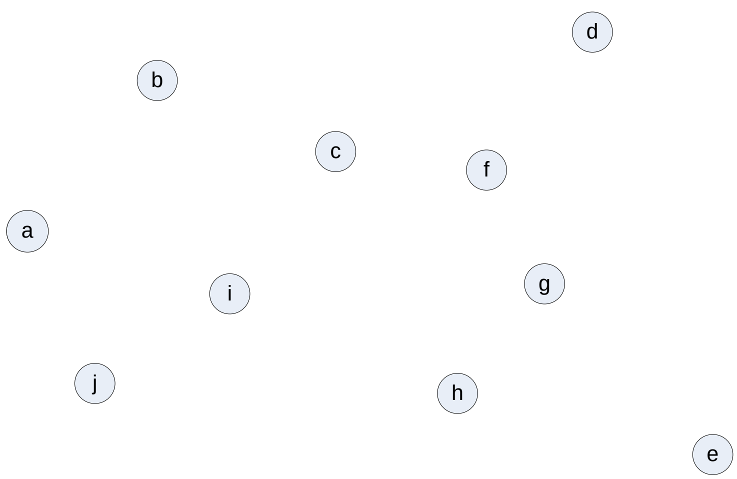


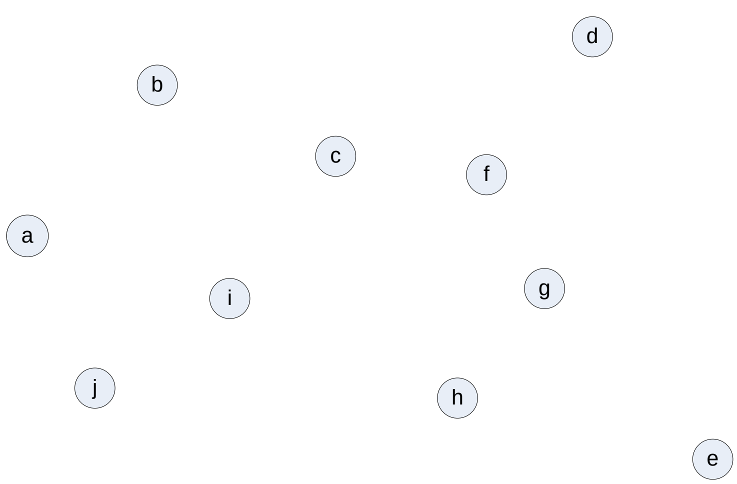


Name:	Matrikelnummer:

Graph:	Heap:







Name:	Matrikelnummer:

AUFGABE 3: Mehrprozessor-Scheduling (8 Punkte)
Entwerfen Sie einen Algorithmus, der n Jobs $(j_1, ..., j_n)$ mit Laufzeiten $(t_1, ..., t_n)$ auf m Arbeitsstationen verteilt.

1. Beweisen oder widerlegen Sie die Optimalität nachfolgender Strategie. Verteile die Jobs auf die m Arbeitsmaschinen wie folgt: zum Zeitpunkt i wird j_i der Maschine $i\ mod\ m$ zugewiesen. Hierbei ist der Wert $\max\limits_{1 \leq k \leq m} l_k$, mit $l_k := Gesamtlast\ von\ Arbeitsstation\ k$, zu minimieren.

2. Sortiere alle Jobs gemäß ihrer Arbeitsdauer absteigend, so dass gilt: $t_{\pi(1)} \geq t_{\pi(2)} \geq ... \geq t_{\pi(n)}$ mit $\pi(i) := i - tes\ Element\ aus\ der\ Permutation\ \pi(1, ..., n)$. Weise zum Zeitpunkt i den Job $j_{\pi(i)}$ der Arbeitsstation m_z mit der bis dato geringsten Auslastung l_z $(l_z = \min\limits_{1 \leq k \leq m} l_k)$ zu und streiche $j_{\pi(i)}$ aus der Liste der Jobs. Zeigen Sie, dass diese Strategie für die in 1.) def. Kostenfunktion 2-approximativ ist, d.h., die so gefundene Lösung nicht schlechter als das 2-fache der optimalen Lösung ist.

Name:	Matrikelnummer:

AUFGABE 4: Hashing (6 Punkte)
Gegeben seien n paarweise verschiedene Zahlen und eine leere Hashtabelle mit statischer Länge m ($m < n$).

a) Sie wollen die Zahlen in die Hashtabelle einfügen. Welche aus der Vorlesung bekannte Technik kann hier zur Kollisionsverwaltung eingesetzt werden, damit alle n Zahlen in die Hashtabelle eingefügt werden können? Begründen Sie Ihre Antwort.

b) Nehmen Sie an, dass alle n Zahlen durch die in a) gewählte Technik in die Hashtabelle eingefügt worden sind. Geben Sie einen Algorithmus mit Laufzeit $O(n \log n)$ an, um alle Zahlen aus der Hashtabelle in absteigender Reihenfolge auszugeben. Sie dürfen dabei eine neue Datenstruktur erstellen und benutzen. Beschreiben Sie Ihren Algorithmus kurz und präzise.

c) Zeigen Sie, dass Ihr Algorithmus die Laufzeitschranke aus b.) einhält.

Name:	Matrikelnummer:

AUFGABE 5: Sortieren (6 Punkte)
Gegeben sei ein Array $A = (a_1, a_2, ..., a_n)$ mit $a_i \in \mathbb{Z}\backslash\{0\}$ für $1 \leq i \leq n$.

a) Entwickeln Sie einen Algorithmus, welcher mittels geeigneter Operationen das Array A in ein Array $A_{Sort} = (a'_1, a'_2, ...a'_j, ..., a'_n)$ überführt, wobei für ein festes j gilt: $a'_i < 0$ für alle $1 \leq i \leq j$ und $a'_i \geq 0$ für alle $j < i \leq n$. Die Laufzeit Ihres Algorithmus sollte $O(n)$ nicht überschreiten. Dabei darf der Algorithmus nur konstant viele zusätzliche Speicherplätze verwenden. Geben Sie den Pseudocode Ihres Algorithmus an.

b) Zeigen Sie die Korrektheit und analysieren Sie das Laufzeitverhalten und den Speicherplatzbedarf Ihres Algorithmus.

Name:	Matrikelnummer:

AUFGABE 6: Längste gemeinsame Teilfolge (6 Punkte)
Berechnen Sie die Länge der längsten gemeinsamen Teilfolge der Folgen

"a a c a b" und "a c c b b"

Verwenden Sie dazu den Algorithmus aus der Vorlesung und füllen Sie die unten abgebildete Tabelle aus. Geben Sie auch die Länge der längsten gemeinsamen Teilfolge an.

	j	0	1	2	3	4	5
i			a	c	c	b	b
0							
1	a						
2	a						
3	c						
4	a						
5	b						

Länge der Folge: ________

Ersatztabelle:

	j	0	1	2	3	4	5
i			a	c	c	b	b
0							
1	a						
2	a						
3	c						
4	a						
5	b						

Länge der Folge: ________

Name:	Matrikelnummer:

AUFGABE 7: Gierige Algorithmen (8 Punkte)
Gegeben seien n Fruchtsäfte $F_1, \ldots, F_n$. Der Fruchtsaft F_i hat dabei den Brennwert von B_i Kalorien pro Liter. Von dem Fruchtsaft F_i haben Sie L_i Liter vorrätig.
Sie möchten aus den vorrätigen Fruchtsäften nun X Liter eines Getränks mit minimalem Brennwert mischen. Der Geschmack des Getränks ist dabei nicht von Interesse. Gesucht ist ein Algorithmus mit Laufzeit $\mathcal{O}(n \log n)$, der die verwendeten Mengen der Fruchtsäfte liefert. Sie können davon ausgehen, dass $\sum_{i=1}^{n} L_i \geq X$ gilt.

- Beschreiben Sie den Algorithmus kurz und präzise.
- Begründen Sie, warum Ihr Algorithmus die geforderte Laufzeitschranke einhält.

Name:	Matrikelnummer:

AUFGABE 8: Datenstrukturen (7 Punkte)
Auf einem Bahnhof hängt ein Monitor zur Anzeige der Nummer des nächsten erwarteten Zugs und seiner Ankunftszeit. Das System zur Verwaltung der Anzeige unterstützt das Einfügen neuer Daten (Zugnummer + Ankunftszeit), bzw. das Löschen und Abrufen von Daten aus der verwalteten Datenmenge. Sie dürfen davon ausgehen, dass über Nacht (23:00-6:00 Uhr) keine Daten im System vorhanden sind und auf dem Monitor nichts angezeigt wird. Die Ankunftszeiten liegen zwischen 6:00 - 23:00 Uhr und das System verwaltet nur Züge, die bis 23:00 Uhr am selben Tag eintreffen.

1. Beschreiben Sie eine Datenstruktur, die die eingehenden Daten gegeben durch das Objekt

   ```
   Zug {
       int Ankunftszeit;
       int Zugnummer;
   }
   ```

   speichert sowie den Zugriff auf die Daten des nächsten erwarteten Zugs erlaubt. Geben Sie folgende Operationen an:

   - Eine Einfüge-Operation, die neue Daten einfügt.
   - Eine Lösch-Operation, die den nächsten erwarteten Zug löscht.
   - Eine Lösch-Operation, die eine Zugnummer entgegen nimmt und diesen Zug mit seiner Ankunftszeit aus der Datenstruktur löscht.
   - Eine First-Operation, die den Zugriff auf die Daten des nächsten erwarteten Zugs erlaubt.

   Beachten Sie dabei, dass die First-Operation eine Laufzeit $O(1)$ und alle anderen Operationen eine Laufzeit $O(\log n)$ haben müssen.

2. Begründen Sie für jede Ihrer Operationen warum diese die geforderte Laufzeit einhält.

Name:	Matrikelnummer:

Name:	Matrikelnummer: