Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Abteilung Algorithmik

Wintersemester 2015/2016

Prof. Dr. Sándor P. Fekete
Dr. Christian Scheffer
Jan-Marc Reinhardt

Klausur
Algorithmen und Datenstrukturen
02. 03. 2016

Name: ....

Vorname: ....

Matr.-Nr.: ....

Studiengang: ....

☐ Bachelor ☐ Master ☐ Andere

Mit der Veröffentlichung meines Klausurergebnisses unter meiner Matrikelnummer bin ich einverstanden.

....
Unterschrift

Hinweise:

- Bitte das Deckblatt vollständig ausfüllen.
- Die Klausur besteht aus 15 Blättern, bitte auf Vollständigkeit überprüfen.
- Erlaubte Hilfsmittel: Keine.
- Eigenes Papier ist nicht erlaubt.
- Die Rückseiten der Blätter dürfen beschrieben werden.
- Die Heftung der Blätter darf nicht entfernt werden.
- Antworten die *nicht* gewertet werden sollen bitte deutlich durchstreichen. Kein Tippex verwenden.
- Mit *Bleistift* oder in *Rot* geschriebene Klausurteile können nicht gewertet werden.
- Werden mehrere Antworten gegeben, werten wir die mit der geringsten Punktzahl.
- Sämtliche Algorithmen, Datenstrukturen, Sätze und Begriffe beziehen sich, sofern nicht explizit anders angegeben, auf die in der Vorlesung vorgestellte Variante.
- Die Bearbeitungszeit für die Klausur beträgt 120 Minuten.

Aufgabe	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	Σ
Punkte	**16**	**10**	**17**	**17**	**10**	**10**	**10**	**10**	**100**
Erreicht									
Note	—	—	—	—	—	—	—	—	

Aufgabe 1: Graphen **(8+4+4 Punkte)**

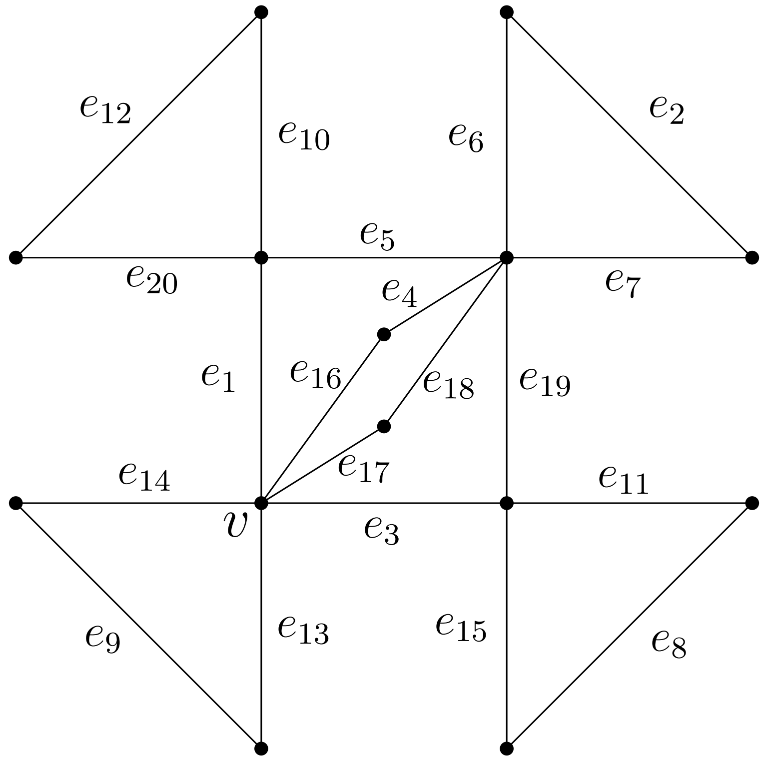


Abbildung 1: Der Graph G

a) Wende Fleurys Algorithmus auf den Graphen G aus Abbildung 1 an. Starte am Knoten v und wähle in jedem Schritt, in dem mehrere Kanten zur Auswahl stehen, diejenige mit dem kleinsten Index. Gib die Reihenfolge der gewählten Kanten an.

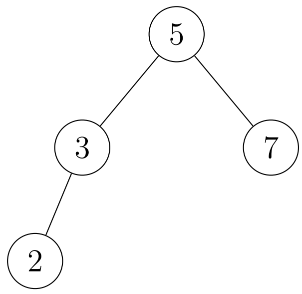
b) Zeige oder widerlege: Jeder einfache, zusammenhängende Graph mit genau vier Knoten, der keinen Eulerweg enthält, enthält auch keinen Hamiltonkreis.

c) Sei $P = (v_0, v_1, v_2, \ldots, v_{k-1}, v_k)$ ein kürzester Weg zwischen v_0 und v_k in einem Graphen G, $k \geq 2$. Beweise: Der Weg $P' = (v_i, v_{i+1}, \ldots, v_{j-1}, v_j)$ ist ein kürzester Weg zwischen v_i und v_j für alle i, j mit $0 \leq i < j \leq k$.

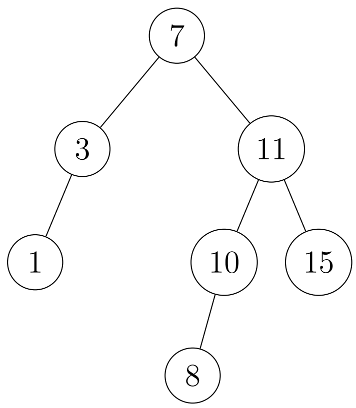
Aufgabe 2: AVL-Bäume **(10 Punkte)**

Betrachte in den Aufgabenteilen a) bis e) den Baum, der in der jeweiligen Abbildung dargestellt wird. Führe die Operation, die in dem jeweiligen Aufgabenteil genannt ist, und die damit verbundenen Restrukturierungsmaßnahmen (damit sind die Algorithmen aus der Vorlesung gemeint, die die AVL-Eigenschaft erhalten) auf dem entsprechenden Baum aus. Zeichne dabei das Resultat nach jeder einzelnen ausgeführten Operation Insert, Delete und Restructure in einen separaten Baum:

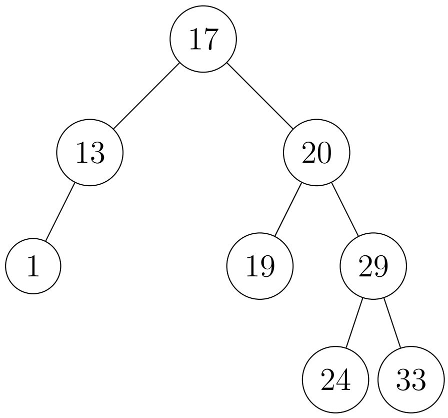
a) Insert$(T, 1)$



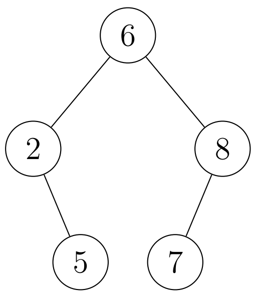
b) Delete$(T, 7)$



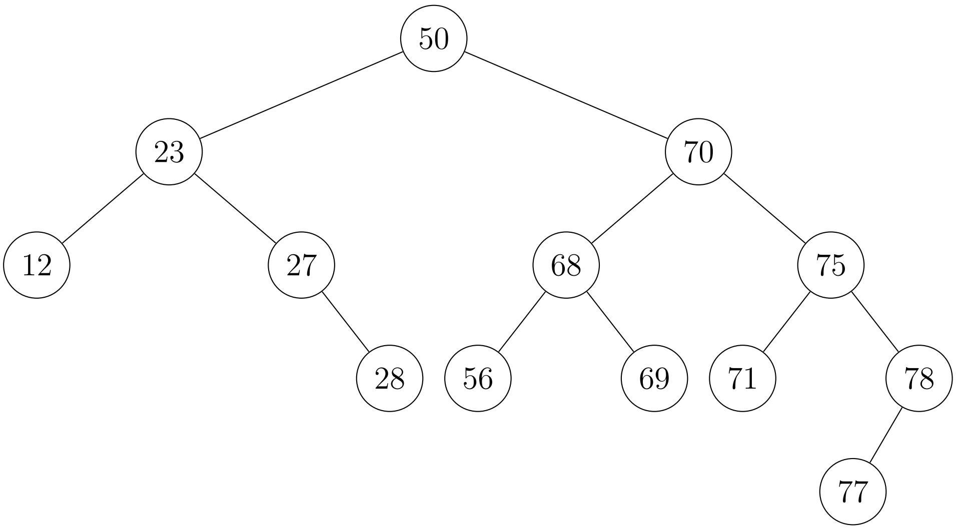
c) $\textsc{Insert}(T, 27)$



d) $\textsc{Insert}(T, 4)$



e) $\textsc{Delete}(T, 12)$



Aufgabe 3: Komplexität **(4+4+5+4 Punkte)**

Seien $f, g, h : \mathbb{N} \to \mathbb{R}^+$ drei Funktionen. $f \cdot g$ ist definiert als $(f \cdot g)(n) = f(n)g(n)$.

a) Zeige oder widerlege: $f \in \mathrm{O}(h) \wedge g \in \mathrm{O}(h) \Rightarrow f \cdot g \in \mathrm{O}(h^2)$

b) Zeige oder widerlege: $f \in \Theta(g) \wedge h \in \mathrm{O}(g) \Rightarrow f \in \mathrm{O}(h)$

c) Fülle die folgende Tabelle aus, indem du einträgst, in welcher Beziehung die in der Zeile angegebene Klasse A zu der in der Spalte angegebenen Klasse B steht. Trage folgende Symbole ein:

- $\subset$, wenn A eine echte Teilmenge von B ist,
- $\supset$, wenn B eine echte Teilmenge von A ist,
- $=$, wenn A und B gleich sind,
- x, wenn weder A eine Teilmenge von B noch B eine Teilmenge von A ist.

Die mit — markierten Zellen müssen nicht ausgefüllt werden.

	$\mathrm{O}(n^3)$	$\Theta(n^2)$	$\Omega(n)$	$\Omega(\log n)$	$\mathrm{O}(n^2)$
$\mathrm{O}(n^3)$	—				
$\Theta(n^2)$	—	—			
$\Omega(n)$	—	—	—		
$\Omega(\log n)$	—	—	—	—	
$\mathrm{O}(n^2)$	—	—	—	—	—

d) Gesucht ist ein Algorithmus, der auf n Elementen arbeitet. Dazu gibt es zwei Möglichkeiten.

(i) Algorithmus 1 betrachtet alle (ungeordneten) Paare von Elementen. Die Bearbeitung jedes Paares geht in $\mathrm{O}(1)$ Zeit.

(ii) Algorithmus 2 benutzt Divide & Conquer und teilt die n Elemente in zwei gleichgroße Teilmengen, auf denen Algorithmus 2 dann rekursiv aufgerufen wird. Aufteilen und Zusammenfügen der Teillösungen geht jeweils in $\mathrm{O}(n)$.

Welche Laufzeiten haben Algorithmus 1 und 2 (jeweils mit Begründung!)? Welches Verfahren ist asymptotisch schneller?

Aufgabe 4: Rekursionen **(4+4+4+5 Punkte)**

a) Wie lautet das Mastertheorem aus der Vorlesung?

b) Bestimme mit Hilfe des Mastertheorems das asymptotische Wachstum der Rekursion

$$T(n) = 3T\left(\frac{n}{2}\right) + 17n^2 + 4T\left(\frac{n}{4}\right)$$

Bestimme die Werte aller im Mastertheorem auftretender Parameter.

c) Betrachte die folgende Rekursionsgleichung:

$$L_0 = 2, L_1 = 1, L_n = L_{n-1} + 2L_{n-2}$$

Berechne L_2 bis L_9 und trage sie in die Tabelle ein.

n	0	1	2	3	4	5	6	7	8	9
L_n	2	1								

d) Bestimme eine geschlossene Form (d.h. eine Funktion in Abhängigkeit von n, die keine Rekursion benutzt) für die Rekursionsgleichung aus c) und beweise ihre Korrektheit per vollständiger Induktion. (Hinweis: Die geschlossene Form muss nicht hergeleitet werden. Du sollst anhand der Werte aus c) eine Vermutung aufstellen und diese per Induktion beweisen.)

Aufgabe 5: Sortieren **(10 Punkte)**

Wende Mergesort auf das Array

$$A = [5, 3, 6, 7, 2, 11, 8, 4, 9, 10, 1]$$

an. Gib dazu separat und in chronologischer Reihenfolge die Ergebnisse aller Mergeschritte auf Teilarrays der Länge ≥ 2 an, indem Du die Zeilen der Tabelle ausfüllst.

$A =$	5	3	6	7	2	11	8	4	9	10	1
1. $A =$											
2. $A =$											
3. $A =$											
4. $A =$											
5. $A =$											
6. $A =$											
7. $A =$											
8. $A =$											
9. $A =$											
10. $A =$											

Aufgabe 6: Median **(10 Punkte)**

Gegeben sei die Menge $X = \{x_1, \ldots, x_n\}$ von paarweise verschiedenen natürlichen Zahlen für $n \in \mathbb{N}$ und $k \in \mathbb{N}$ mit $1 \leq k \leq n$.

a) Wie ist das *k-te Element von X* definiert?

b) Beschreibe in eigenen Worten das Vorgehen des Select-Algorithmus.

c) Welche Laufzeit $T(n)$ hat der SELECT-Algorithmus? Gib die rekursive Darstellung von $T(n)$ an und erkläre diese.

d) Welche Auswirkung hat es auf X_2, dass X nur paarweise verschiedene Zahlen enthält?

e) Wende den SELECT-Algorithmus auf $X = \{4, 2, 10, 5, 9, 7, 14, 3, 15, 1, 6, 8, 11, 12, 13\}$ und $k = 3$ an. Gib hierbei nur für die oberste Rekursionsebene die berechneten Blöcke, M' und m an, indem Du das folgende Ablaufdiagramm komplettierst. Markiere die Teilmenge von X, in der das k-te Element rekursiv weiter gesucht wird.

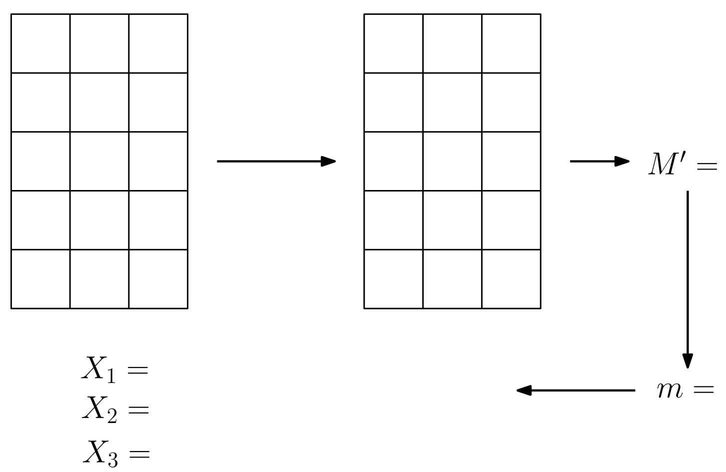


Abbildung 2: Ablaufdiagramm von SELECT für die oberste Rekursionsebene.

Aufgabe 7: Algorithmenentwurf **(4+6 Punkte)**

Gegeben seien ein Array A[1,...,n], das positive ganze Zahlen enthält, und eine Zahl k. Die Frage ist, ob A zwei Zahlen enthält, deren Summe k ergibt. Ein Algorithmus, der das Problem löst, gibt `true` zurück, wenn es $i \neq j$ mit A[i]+A[j] $= k$ gibt, sonst `false`.

a) Formuliere einen Algorithmus, der das Problem in $\Theta(n^2)$ löst.

b) Formuliere einen Algorithmus, der das Problem in $O(n \log n)$ löst. Für diese Teilaufgabe darfst du jeden in der Vorlesung vorgestellten Algorithmus als Subroutine verwenden.

Aufgabe 8: Kurzfragen **(2+2+2+2+2 Punkte)**

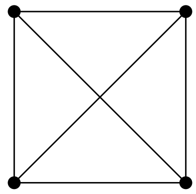


Abbildung 3: Eine Darstellung des K_4

a) Nach dem Löschen eines Elements aus einem AVL-Baum reicht immer ein `Restructure`, um den Baum zu balancieren. ☐ wahr ☐ falsch

b) Hamiltonpfade benutzen jede Kante höchstens einmal. ☐ wahr ☐ falsch

c) Jede Permutation der Zahlen $1, 2, \ldots, n$ kann in $\mathrm{O}(n)$ sortiert werden. ☐ wahr ☐ falsch

d) Der K_4 ist planar. ☐ wahr ☐ falsch

e) Das Bauen eines Max-Heaps mit n Knoten benötigt mindestens $\Omega(n \log n)$ Zeit. ☐ wahr ☐ falsch

Viel Erfolg ☺