/ 16 P

Aufgabe 1.

Instruktionen:

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) Führen Sie auf dem folgenden Array zwei Iterationen des Sortieralgorithmus *Sortieren durch Einfügen* aus. Das zu sortierende Array ist durch vorherige Iterationen bereits bis zum Doppelstrich sortiert worden.

3	7	10	15	8	6	9	5	2	13
1	2	3	4	5	6	7	8	9	10

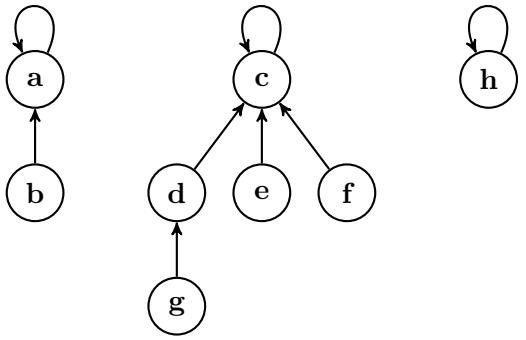
1	2	3	4	5	6	7	8	9	10

1	2	3	4	5	6	7	8	9	10

/ 1 P b) Fügen Sie die Schlüssel 20, 24, 31 und 7 in dieser Reihenfolge in die untenstehende Hashtabelle ein. Benutzen Sie offenes Hashing mit der Hashfunktion $h(k) = k \bmod 11$. Lösen Sie Kollisionen mittels quadratischem Sondieren auf. Im Falle einer Kollision soll die Sondierung *zunächst nach links und erst danach nach rechts* erfolgen.

11				15	5			19	9	
0	1	2	3	4	5	6	7	8	9	10

/ 1 P c) Führen Sie auf der folgenden Union-Find-Datenstruktur $\textsc{Union}(h, \textsc{Find}(g))$ mithilfe des Verfahrens "Vereinigung nach Höhe" aus und zeichnen Sie die resultierende Datenstruktur nach der Operation.



Nach der Operation:

/ 1 P d) Das folgende Array enthält die Elemente eines in üblicher Form gespeicherten Min-Heaps. Entfernen Sie das minimale Element aus dem Heap, stellen Sie die Heap-Bedingung wieder her, und geben Sie das resultierende Array an.

1	3	6	4	9	7	10	13	8
1	2	3	4	5	6	7	8	9

1	2	3	4	5	6	7	8

/ 1 P e) Geben Sie eine kürzeste Folge von Anfragen an, welche ausgehend von der Liste

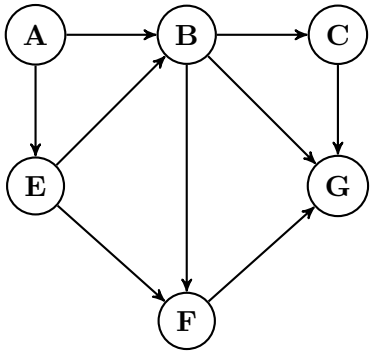
$$A \to B \to C \to D \to E \to F$$

unter Anwendung der Move-to-Front-Regel folgende Liste ergibt:

$$D \to A \to C \to B \to E \to F$$

Anfragen: _____

/ 1 P f) Der folgende Graph wird ausgehend von Knoten A traversiert. Die Nachbarn eines Knoten werden in alphabetischer Reihenfolge abgearbeitet. Geben Sie die beiden Reihenfolgen an, in der eine Tiefensuche und eine Breitensuche die Knoten inspizieren.



Tiefensuche:

A, ———, ———, ———, ———, ———.

Breitensuche:

A, ———, ———, ———, ———, ———.

/ 1 P g) Zeichnen Sie den natürlichen Suchbaum T, der entsteht, wenn die Schlüssel $5, 4, 8, 7, 1, 6, 10$ in dieser Reihenfolge in einen anfangs leeren natürlichen Suchbaum eingefügt werden. Geben Sie die Preorder-, Postorder- und Inorder-Reihenfolgen der Schlüssel in T an.

natürlicher Suchbaum T:

Preorder-Reihenfolge:

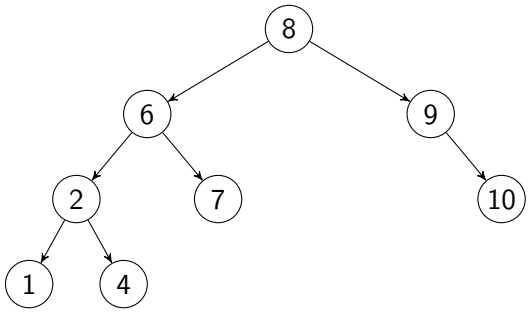
Inorder-Reihenfolge:

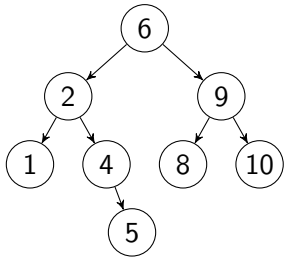
Postorder-Reihenfolge:

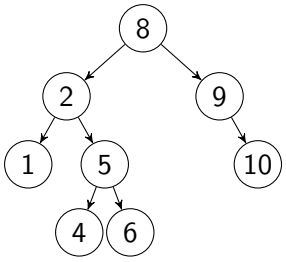
/ 2 P h) Kreuzen Sie bei jeder der folgenden Datenstrukturen an, ob der darin verwendete Baum balanciert ist (d.h. eine Höhe logarithmisch in der Anzahl der gespeicherten Schlüssel garantiert) oder nicht balanciert ist. Jede korrekte Antwort gibt 0.5 Punkte, für jede falsche Antwort werden 0.5 Punkte abgezogen. Eine fehlende Antwort gibt 0 Punkte. Insgesamt gibt die Aufgabe mindestens 0 Punkte. Sie müssen Ihre Antworten nicht begründen.

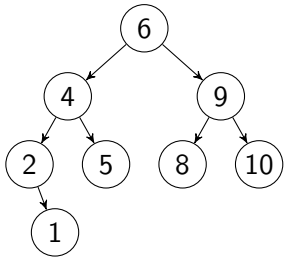
B-Baum	☐ BALANCIERT	☐ NICHT BALANCIERT
Heap	☐ BALANCIERT	☐ NICHT BALANCIERT
AVL-Baum	☐ BALANCIERT	☐ NICHT BALANCIERT
Natürlicher Suchbaum	☐ BALANCIERT	☐ NICHT BALANCIERT

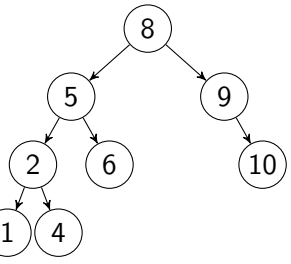
/ 1 P i) Welcher AVL-Baum entsteht, wenn im folgenden AVL-Baum zuerst Schlüssel 7 gelöscht wird und anschliessend Schlüssel 5 eingefügt wird?

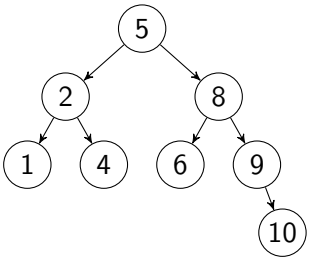


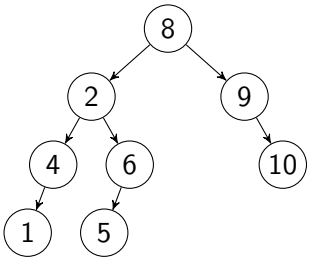
☐ 

☐ 

☐ 

☐ 

☐ 

☐ 

/ 3 P j) Gegeben ist die folgende Rekursionsgleichung:

$$T(n) := \begin{cases} 5 \cdot T(n/8) + 8 & n > 1 \\ 1 & n = 1 \end{cases}$$

Geben Sie eine geschlossene (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 8 ist.
Benutzen Sie also $n = 8^k$ oder $k = \log_8(n)$.

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

Herleitung (falls benötigt):

Geschlossene und vereinfachte Formel:

$T(n) = T(8^k) =$

Induktionsbeweis:

Induktionsbeweis (Fortsetzung):

/ 1 P k) Geben Sie die asymptotische Laufzeit in Abhängigkeit von $n \in \mathbb{N}$ für den folgenden Algorithmus (so knapp wie möglich) in Θ-Notation an. Sie müssen Ihre Antwort nicht begründen.

```
1 for ( int i = n; i >= 1; i = i - 5 ) {
2         int j = n;
3         while ( j > 1 ) {
4                 j = j / 2;
5         }
6 }
```

Laufzeit in möglichst knapper Θ-Notation:

/ 1 P l) Geben Sie die asymptotische Laufzeit in Abhängigkeit von $n \in \mathbb{N}$ für den folgenden Algorithmus (so knapp wie möglich) in Θ-Notation an. Sie müssen Ihre Antwort nicht begründen.

```
1 for ( int i = 1; i <= n; i = i + 1 ) {
2         for ( int j = 1; j < 2 * i; j = j + 2) {
3                 ;
4         }
5 }
```

Laufzeit in möglichst knapper Θ-Notation:

/ 1 P m) 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.

$$\frac{n^2}{\log(n)},\ \log(n^2),\ \sqrt{6^n},\ 2^{15},\ n\sqrt{n},\ \binom{n}{6},\ n\log(n)$$

Lösung: _____, _____, _____, _____, _____, _____, _____.

Aufgabe 2. **/ 10 P**

Ein *Palindrom* ist eine Zeichenfolge, die sich von vorne wie von hinten gleich liest, also z.B. das Wort `RENTNER`. Formal ist ein Palindrom eine Zeichenfolge der Länge 0 oder 1, oder eine Zeichenfolge $\langle a_1, \ldots, a_l \rangle$, wobei $a_1 = a_l$ gilt und $\langle a_2, \ldots, a_{l-1} \rangle$ ein Palindrom ist.

Gegeben sei eine Zeichenfolge $Z = \langle a_1, \ldots, a_n \rangle$ der Länge n. Wir sagen, dass Z ein Palindrom P enthält, wenn P eine (nicht notwendigerweise zusammenhängende) Teilfolge von Z ist. Gesucht ist das längste in Z enthaltene Palindrom.

Beispiel: Die Zeichenfolge $Z = \langle \mathtt{A}, \mathtt{N}, \mathtt{A}, \mathtt{N}, \mathtt{A}, \mathtt{S} \rangle$ enthält das Palindrom der Länge 0, sowie die Palindrome `A`, `N`, `S`, `AA`, `NN`, `AAA`, `ANA`, `NAN`, `ANNA` sowie `ANANA` (einige dieser Palindrome kommen mehrfach vor). Das längste in Z enthaltene Palindrom ist also `ANANA`.

Entwerfen Sie für das obige Problem einen Algorithmus, der nach dem Prinzip der *dynamischen Programmierung* arbeitet. Gehen Sie dabei auf die folgenden Aspekte ein.

1) Was ist die Bedeutung eines Tabelleneintrags, und welche Grösse hat die DP-Tabelle?
2) Wie kann die Tabelle initialisiert werden, und 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 auslesen?

Geben Sie auch die Laufzeit Ihrer Lösung an, und begründen Sie Ihre Antwort.

Grösse der DP-Tabelle / Anzahl Einträge: _____

Bedeutung eines Tabelleneintrags:

$DP[\_\_\_\_\_]$: _____

_____

Berechnung eines Eintrags:

$DP[\_\_\_\_\_] =$

Berechnungsreihenfolge:

Auslesen des längsten enthaltenen Palindroms:

Laufzeit (mit Begründung):

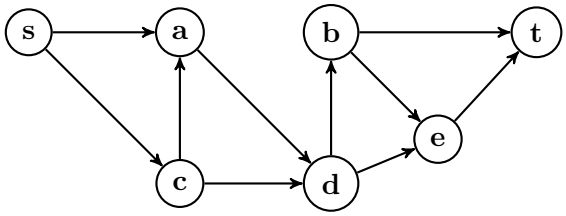
/ 7 P

Aufgabe 3.

Zwei Computer sollen über ein Netz zuverlässig kommunizieren. Die Computer sind nicht unbedingt direkt miteinander verbunden, sondern schicken im Allgemeinen Pakete über verschiedene Zwischenstationen durchs Netz. Es soll sichergestellt werden, dass ein Computer dem anderen Computer auch bei einem Ausfall einer einzigen, beliebigen Verbindung im Netz Pakete schicken kann.

Das Netz ist durch eine Menge V von Knoten und eine Menge E von (gerichteten) Verbindungen zwischen je zwei Knoten gegeben. Weiterhin sind zwei Knoten s und t in V gegeben, die miteinander kommunizieren sollen.

/ 3 P a) Wir wollen entscheiden, ob s auch dann noch Pakete an t schicken kann, wenn eine einzige, beliebige Verbindung in E unterbrochen wird. Modellieren Sie dazu das Problem als Flussproblem. Beschreiben Sie dazu die Konstruktion eines geeigneten Netzes $N_a = (V_a, E_a, c_a)$ und geben Sie an, welche Kapazitäten c_a die Kanten besitzen sollen. Wie kann aus dem Wert eines maximalen Flusses abgelesen werden, ob die Kommunikation von s nach t möglich ist, wenn eine einzige, beliebige Verbindung in E unterbrochen wird?



Beispiel: Im links dargestellten Netz kann der Knoten s auch dann noch Pakete an den Knoten t schicken, wenn eine einzige, beliebige Kante entfernt wird.

/ 4 P b) Jetzt möchten wir entscheiden, ob der Knoten s auch dann noch Pakete an den Knoten t schicken kann, wenn ein *Knoten* aus $V \setminus \{s, t\}$ defekt ist und nicht genutzt werden kann. Modellieren Sie dazu das Problem als Flussproblem und beschreiben Sie ein geeignetes Netz $N_b = (V_b, E_b, c_b)$. Wie kann aus dem Wert eines maximalen Flusses in diesem Netz abgelesen werden, ob die Kommunikation von s nach t möglich ist, wenn nur ein Knoten entfernt wird?

Beispiel: Im oben dargestellten Netz kann der Knoten s keine Pakete an den Knoten t schicken, wenn der Knoten d nicht genutzt werden kann.

Teilaufgabe a)

Definition des Netzes N_a (wenn möglich in Worten und nicht formal):

Knotenmenge V_a: _____

_____

Kantenmenge E_a: _____

_____

Kapazitäten c_a: _____

_____

Schematische Darstellung / Zeichnung des Netzes:

Eine Kommunikation von s nach t ist möglich, falls: _____

_____

Teilaufgabe b)

Definition des Netzes N_b (wenn möglich in Worten und nicht formal):

Knotenmenge V_b: _____

_____

Kantenmenge E_b: _____

_____

Kapazitäten c_b: _____

_____

Schematische Darstellung / Zeichnung des Netzes:

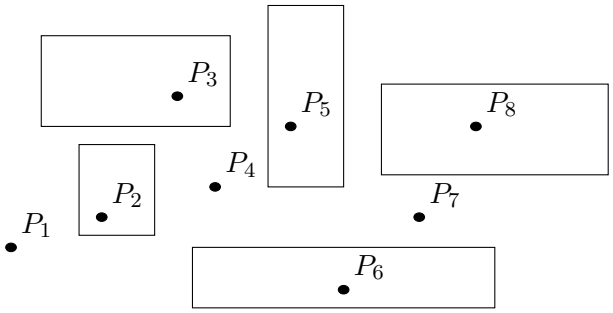
Eine Kommunikation von s nach t ist möglich, falls: _____

_____

Aufgabe 4. **/ 9 P**

Gegeben sei eine Menge von n Punkten $P_1, \ldots, P_n$ mit $P_i = (x_i, y_i) \in \mathbb{Q}^2$ für alle $i \in \{1, \ldots, n\}$. Keine zwei Punkte haben dieselben x- oder y-Koordinaten. Gegeben sei ausserdem eine Menge von m achsen-parallelen Rechtecken $R_1, \ldots, R_m$ in der Ebene. Jedes Rechteck R_j mit $j \in \{1, \ldots, m\}$ ist durch seinen linken, unteren Eckpunkt $(l_j, u_j) \in \mathbb{Q}^2$ und seinen rechten, oberen Eckpunkt $(r_j, o_j) \in \mathbb{Q}^2$ gegeben. Keine zwei Rechtecke überlappen oder berühren sich, und kein Punkt P_i liegt genau auf dem Rand eines Rechtecks. Gesucht sind alle Punkte, die sich nicht innerhalb eines der m Rechtecke befinden.

Beispiel: In der Abbildung rechts sind fünf Rechtecke und acht Punkte gegeben. Die Punkte P_1, P_4 und P_7 befinden sich in keinem Rechteck.



Entwerfen Sie einen möglichst effizienten Scanline-Algorithmus für das obige Problem. Gehen Sie in Ihrer Lösung auf die folgenden Aspekte ein.

1) In welche Richtung läuft die Scanline, und was sind die Haltepunkte?
2) Welche Objekte verwaltet die Scanline-Datenstruktur, und was ist eine angemessene Datenstruktur?
3) Was passiert, wenn die Scanline auf einen neuen Haltepunkt trifft?
4) Wie können diejenigen Punkte bestimmt werden, die sich nicht innerhalb eines Rechteckes befinden?
5) Welche Laufzeit in Abhängigkeit von n und m hat Ihr Algorithmus? Begründen Sie Ihre Antwort.

Scanline-Richtung (genau eine Möglichkeit ankreuzen):

- Die Scanline ist eine *vertikale Gerade*; sie bewegt sich
  ☐ von links nach rechts / ☐ von rechts nach links.
- Die Scanline ist eine *horizontale Gerade*; sie bewegt sich
  ☐ von oben nach unten / ☐ von unten nach oben.
- Die Scanline ist eine *Halbgerade*; sie rotiert
  ☐ im Uhrzeigersinn / ☐ im Gegenuhrzeigersinn um den Punkt _____.

☐ andere Richtung: _____

_____.

Haltepunkte:

Menge der Haltepunkte: _____

_____

sortiert nach: _____

_____

Scanline-Datenstruktur:

Operationen bei einem Haltepunkt:

Auslesen der Lösung:

Laufzeit des Algorithmus: