1.

Geben Sie eine genaue Definition der Klasse NP an. (6 Punkte)

✗ 0

2. (18 Punkte)

Gegeben sei der Graph $G = (V, E)$ mit

- $V = \{1, 2, 3, 4, 5, 6, 7\}$
- $E = \{(1,7), (7,5), (2,1), (2,5), (4,2), (4,1), (1,4), (5,6), (7,4)\}$

(a) Handelt es sich bei G um einen gerichteten Graphen?

(b) Geben Sie die Adjazenzmatrix von G an.

(c) Stellen Sie G mit Hilfe von Adjazenzlisten dar.

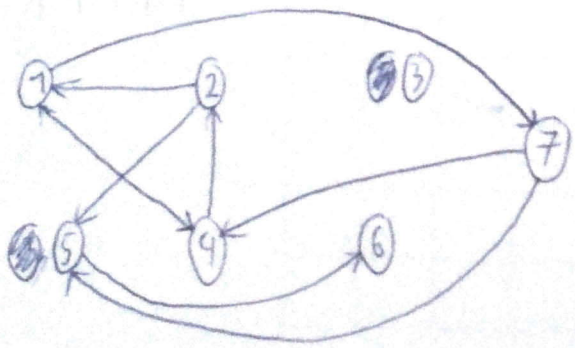
(d) Geben Sie eine mögliche Reihenfolge an, in der die Knoten bei einer Breitensuche in die dabei verwendete Queue eingefügt würden, vorausgesetzt man startet die Suche bei Knoten 4.

(e) Geben Sie eine mögliche Reihenfolge an, in der die Knoten bei einer Tiefensuche besucht würden, wenn als erstes Knoten 4 besucht wird.

a. Ja, es handelt sich um einen gerichteten Graphen ✓ 2/2

b.
$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$
✓ 4/4

c.



kein Adj.-Liste

0/4

d. ✗ 0/4

e. 4 → 2 → 1 → 7 → 5 → 6 → 3

3/4

3.

(8 Punkte)

Sie sehen unten die Darstellung eines unsortierten Arrays von Zahlen. Sortieren Sie dieses Array mit Hilfe des Quicksort-Algorithmus. Halten Sie dabei jede Änderung, die im Array vorgenommen wird (z.B. wenn zwei Zahlen vertauscht werden) in einem eigenen Zwischenschritt fest.

0	1	2	3	4	5	6
7	8	3	1	11	5	9
~~7~~	~~8~~	~~3~~	~~1~~	~~11~~		
7	8	5	1	11	3	9
1	8	5	7	11	3	9
1	3	5	7	11	8	9
1	3	5	7	8	11	9
1	3	5	7	8	9	11

?

Wie geht die Auswahl des Pivot-Elements vor sich?

Wie ändert es am seinen Platz?

2/8

4.

(12 Punkte)

Bestimmen Sie die asymptotische Laufzeit des untigen Algorithmus bei Aufruf der Funktion `f` im schlechtesten Fall (Sie können dabei die O-Notation verwenden). Bestimmen Sie dazu die asymptotische Laufzeit aller unten aufgeführten Funktionen im schlechtesten Fall. Notieren Sie die Laufzeit jeweils als Kommentar am Code.

Eingabe von `f`: Ein Array von Zahlen `a` und eine Zahl `n`. Dabei gilt, dass `a` die Größe `n` besitzt.

```
//Laufzeit: 5n+1
function f(int[] a, int n) {
    for (int i=0; i<n; i++) {
        onTestExchange(a, n, i, 2*i)
        onTestExchange(a, n, i, 2*i+1)
    }
    return a;
}

// Laufzeit: 5n
function onTestExchange(int[] a, int n, int x, int y) {
    if (y < n) and (a[y] < a[x]) {
        exchange(a, x, y);
        upPropagate(a, x);
    }
}

//Laufzeit: 7 Takte
function exchange(int a[], int i, int j) {
    int x = a[i];
    a[i] = a[j];
    a[j] = x;
}

//Laufzeit: 6n
function upPropagate(int[] a, int position) {
    if (position > 0) {
        int father = position/2;
        if (a[father] > a[position]) {
            exchange(a, father, position);
            upPropagate(a, father);
        }
    }
}
```

Kostet nichts?

Kostet nichts?

2/12

5. (12 Punkte)

Gegeben sei die folgende Liste mit Aussagen. Bitte kreuzen Sie an, welche der Aussagen wahr sind, und welche falsch.

Beachten Sie: Für jede richtig angekreuzte Antwort erhalten Sie einen Punkt, für jede falsch angekreuzte Anwort bekommen Sie einen Punkt abgezogen. Insgesamt können Sie auf diese Aufgabe aber nicht weniger als 0 Punkte bekommen.

wahr	falsch	Aussage
☐	☒	$O(\log n^2) = O(\log n)$
☐	☐	Sind f, g Funktionen mit $f \leq_{\text{ae}} g$, so ist $g \in O(f)$.
☐	☐	Für alle f, für die $f(n) = O(n^2)$ gilt, gilt auch $f(n) = O(n \cdot \log n)$.
☒	☐	Ist f die Worst-Case Laufzeit eines Algorithmus A und g seine Average-Case Laufzeit, so gilt $g \leq_{\text{ae}} f$.
☒	☐	Ist $c \in \mathbb{R}$ so ist $c \cdot n^2 \neq O(n^2)$
☐	☒	$O(n^2) = O(n^3)$
☐	☒	Jeder vergleichsbasierte Sortieralgorithmus benötigt höchstens Zeit $O(n \cdot \log n)$.
☐	☐	Ist $k \in \mathbb{N}$, so ist $\pi^k \cdot n = O(n)$.
☐	☒	$O(\log_2 n) = O(\log_e n)$
☐	☐	Für jede Funktion f gilt $f(n) = O(2^n)$.
☐	☒	Ist $c \in \mathbb{N}$ so ist $c \cdot n^2 = O(n^2)$
☒	☐	P $\subseteq$ NP

6. (20 Punkte)

(a) Geben Sie einen schnellen Algorithmus an (nicht langsamer als $O(n \cdot \log n)$), um einen Min-Heap aus einem beliebigen Zahlen-Array a zu erzeugen. Sie dürfen dabei auf eine vorhandene Funktion

`heapify(int[] a, int li, int re)`

zurückgreifen, die das Teilarray von `a`, das zwischen `li` und `re` liegt (beide Grenzen einschließlich, `li`≤`re`), in ein Teilarray in Min-Heap-Form umwandelt, vorausgesetzt, das entsprechende Teilarray ist heap-ähnlich. Die Laufzeit von `heapify` beträgt $O(\log(\texttt{re} - \texttt{li}))$. Begründen Sie, warum Ihr Algorithmus ausreichend schnell ist.

(b) Gegeben sei ein Array a mit dem Inhalt $(9, 3, 2, 7, 5, 1, 0)$ in dieser Reihenfolge. Vollziehen Sie Ihren Algorithmus auf Eingabe a nach und zeichnen Sie die Veränderungen in a im Verlauf des Algorithmus auf (beispielsweise wie in Aufgabe 3, oder durch eine Grafik), so dass man schrittweise nachvollziehen kann, welche Veränderungen im Array Ihr Algorithmus bewirkt. Beziehen Sie dabei die Auswirkungen der Funktion `heapify` mit ein, in deren Verlauf ein Element in einem Baum "nach unten driftet". Geben Sie am Ende das Ergebnis Ihres Algorithmus auf Eingabe a an.

7. (16 Punkte)

Gegeben sei die Hashfunktion $h(x) \stackrel{\text{def}}{=} 2x \mod 11$ und die Sondierungsfunktion $s(x,i) \stackrel{\text{def}}{=} h(x)+h(i+1) \mod 11$.

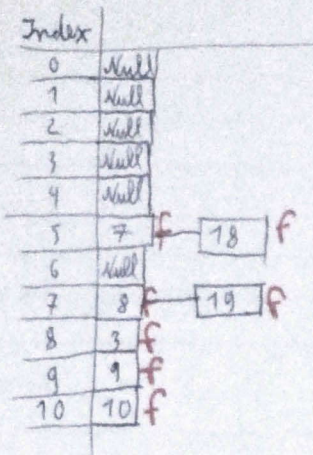
(a) Fügen Sie der Reihe nach die Elemente $3, 8, 19, 7, 18, 9, 10$ in dieser Reihenfolge in ein gehashtes Array der Größe 11 ein. Notieren Sie in der unteren Tabelle die Position, in der das jeweilige Element gespeichert wird.

Index:	0	1	2	3	4	5	6	7	8	9	10
Einfügen der 3 :				3					3		
Einfügen der 8 :				3				8			
Einfügen der 19 :								19			
Einfügen der 7 :						7					
Einfügen der 18 :						18					
Einfügen der 9 :										9	
Einfügen der 10 :											10

f f f f f f f

0/8

(b) Zeichnen Sie eine Grafik, in der die Situation nach Einfügen der selben Zahlen in der selben Reihenfolge in ein gehashtes Array mit der selben Hashfunktion h dargestellt wird, wobei zur Kollisionsbehandlung Überlauflisten verwendet werden.



Ich lege mal zugrunde, dass sie, an welchen Stellen auch immer, die Hashfunktion $h'(x) = 2x+2 \mod 11$ verwenden. Dann läuft sie komplett richtig, außer bei der 10

5/8

8. (8 Punkte)

Zeigen Sie: Sind $A, B \in \mathrm{P}$, so ist auch $A \cap B$ in P.

f o