1. Aufgabe

a)

> Vergleichen Sie das asymptotische Wachstum der Funktionen f und g. Zeigen Sie, ob $f \in \mathcal{O}(g)$ oder $f \in \Omega(g)$ gilt.
>
> 1. $$f(n) = 2^{2n+1}; g(n) = 3^{n+2}$$
> 2. $$f(n) = 6\sqrt{n}(\log n)^2; g(n) = 5n\sqrt{\log(n^5)}$$
> 3. $$f(n) = \sqrt{n^2 \log n^5}; g(n) = 7n\sqrt{n}$$
> 4. $$f(n) = 2^{2n+1}; g(n) = 3^{n+2}$$
> 5. $$f(n) = 5 \cdot 4^n; g(n) = 3^{2n+1}$$

b)

> Bestimmen Sie mit Hilfe einer geeigneten Methode die Größenordnung (Θ) der Lösung folgender Rekurrenzen.
>
> 1. $$T(n) = 8T(\frac{n}{16}) + \sqrt[3]{n^2}$$
> $$T(n) = 2T(\sqrt{n}) + 1$$
> 2. $$T(n) = 10T(\frac{n}{3}) + n\sqrt[4]{n}$$
> $$T(n) = T(\frac{n}{3}) + T(\frac{n}{4}) + n$$
> 3. $$T(n) = 2T(\frac{n}{6(?)}) + \sqrt{n}$$
> $$T(n) = T(\frac{2n}{5}) + T(\frac{3n}{5}) + n$$
> 4. $$T(n) = 8T(\frac{n}{16}) + \sqrt[3]{n^2}$$
> $$T(n) = 2T(\sqrt{n}) + 1$$
> 5. $$T(n) = 32T(\frac{n}{8}) + n^2 \log n$$
> $$T(n) = T(\frac{n}{2}) + T(\frac{n}{4}) + n$$

2. Aufgabe

a)

> Definieren Sie den Begriff des binären Suchbaums?

b)

> Fügen sie die Schlüssel $20, 6, 17, 31, 12, 19, 15, 1, 4$ in dieser Reihenfolge in einen anfangs leeren binären Suchbaum ein. Zeichnen Sie den Baum mindestens nach jeder zweiten Einfügeoperation. Löschen Sie die 6.

c)

> Zeigen Sie, dass der Aufbau eines binären Suchbaums aus einer Folge $X = (x_1, x_2, \ldots, x_n)$ von Schlüsseln mindestens $\Omega(n \log n)$ Zeit erfordert.

d)

> In welcher Zeit kann ein balancierter Suchbaum aus einer aufsteigend sortierten Folge $Y = (y_1, y_2, \ldots, y_n), y_1 \leq y_2 \leq \cdots \leq y_n$ aufgebaut werden?

3. Aufgabe

a)

> Definieren Sie den Begriff des Rot-Schwarz-Baums.

b)

> Was versteht man unter der algorithmischen Aufgabe SELECTION? Geben Sie Input und Output an.

> Wir wollen SELECTION als Operation auf Rot-Schwarz-Bäume ausführen. Dabei soll nach dem Aufbau des Baumes (als Preprocesseing) eine SELECTION-Anfrage in $\mathcal{O}(\log n)$ Zeit bearbeitet werden können.

c)

> Geben Sie einen Algorithmus zur Bearbeitung einer SELECTION-Anfang an und analysieren Sie seine Laufzeit

> Es kommt nicht darauf an, den Algorithmus in Pseudocode aufzuschreiben. Wichtiger ist, die Hauptschritte Ihres Algorithmus kurz und klar verbal zu beschreiben und jeweils eine Aussage über die Laufzeit zu treffen.

d)

> Fügen Sie die Schlüssel $25, 14, 42, 6, 2, 20, 10, 22$ in dieser Reihenfolge in einem angfangs leeren Rot-Schwarz-Baum ein. Wie groß ist die Schwarzhöhe des reslutierenden Baumes?

e)

> Zeichnen Sie einen Rot-Schwarz-Baum, der die Schlüssel $1, 2, 3, 4, 5, 6$ enthält und eine möglichst große Höhe hat.

f)

> Zeichnen Sie einen Rot-Schwarz-Baum, der die Schlüssel $1, 2, 3, 4, 5, 6$ enthält und eine möglichst kleiner Höhe hat.

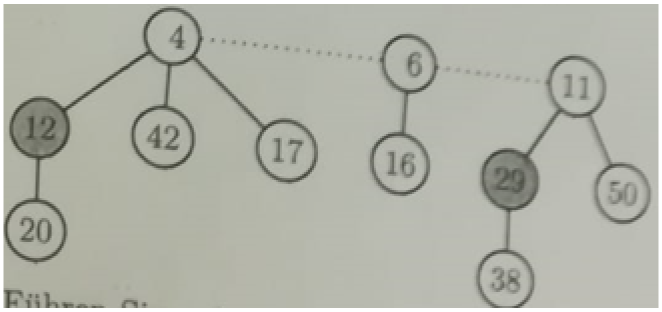
g)

> Die in einem Rot-Schwarz-Baum gespeicherten Schlüssel sollen in sortierter Reihenfolge ausgegeben werden. Geben Sie einen Algorithmus für diese Aufgabe an und begründen Sie kurz, dass die ausgegebene Folge tatsächlich sortiert ist.

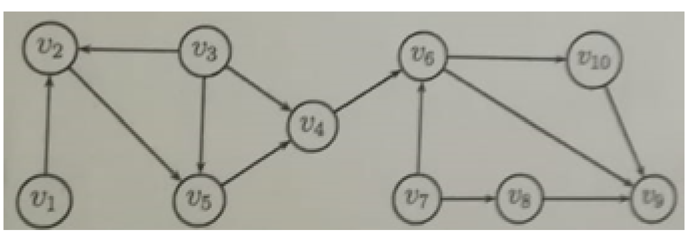
h)

> Zuzeigen:
> Aufbau eines binären Suchbaums aus Folge $X = (x_1, \ldots, x_n)$ dauert mindesten $\Omega(n \log n)$ Zeit.

4. Aufgabe

> Gegeben ist folgender Fibonacci-Heap:
>
> 
>
>
> Führen Sie auf diesem Fibonacci-Heap nacheinander die folgenden Operationen aus. Geben Sie Zwischenschritte an.
>
> - Löschen Sie das Minimum. Geben Sie beim Konsolidieren jeweils das Hilfsfeld an.
> - Setzen Sie den Schlüssel 38 auf 14 herab.
> - Fügen Sie die Schlüssel 49, 22, 35, 15 und 3 ein.
> - Löschen Sie das Minimum. Geben Sie beim Konsolidieren jeweils das Hilfsfeld an.
> - Setzen Sie den Schlüssel 50 auf 33 herab.

5. Aufgabe

> 
>

a)

> Führen Sie den Algorithmus TIEFENSUCHE auf obigem Graphen mit Startknoten v_1 aus. Bestimmen Sie dabei die Kanten des Tiefensuchbaumes und für jeden Knoten u die Werte $d(u)$ und $f(u)$.

b)

> Wie groß ist die Laufzeit von TIEFENSUCHE?

c)

> Was versteht man unter der algorithmischen Aufgabe TOPOLOGISCHES SORTIEREN? Geben Sie Input und Output an.

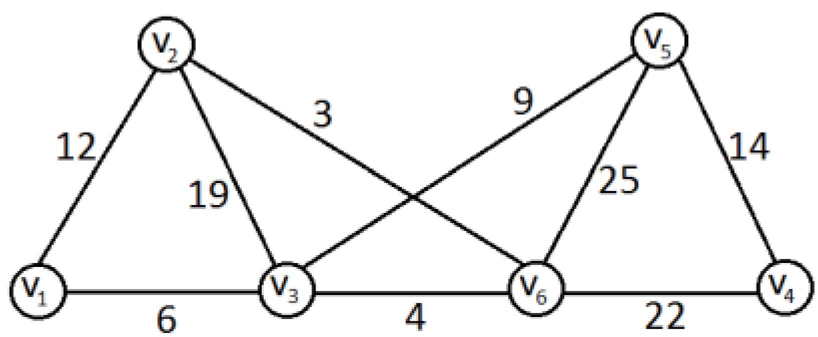
d)

> Wenden Sie TOPOLOGICAL SORT(G) auf den oben gegebenen Graphen an.

e)

> Nennen Sie ein Beispiel für ein algorithmisches Problem, das mit dieser Modellierung gelöst werden kann.

6. Aufgabe



a)

> Was ist der Minimal Spanning Tree (MST) eines kantengewichteten Graphen $G = (V, E)$?

b)

> Welchen abstrakten Datentyp verwendet der Algorithmus von Prim? Wie würden Sie diesen Typ implementieren?

c)

> Wenden Sie den Algorithmus von Prim mit Startknoten v_1 auf den obigen Graphen an. Geben Sie nach Abarbeitung jedes Knotens als Zwischenergebnis den aktuellen Baum und die Elemente in der Datenstruktur zusammen mit ihrem Schlüssel an.

d)

> Beschreiben Sie kurz eine Idee, wie man in $\mathcal{O}(|V|)$ Zeit testen kann, ob ein Graph $G = (V, E)$ (der nicht notwendig zusammenhängend ist) ein Baum ist.

7. Aufgabe

a)

> Es sei $A = [x_1, x_2, \ldots, x_n]$ ein Feld ganzer Zahlen. Definieren Sie den Begriff Rang eines Elements x_i in A und Median von A.

b)

> Wir betrachten $A = [3, -5, 6, 8, 4, 1, -2, 7, 9]$. Geben Sie die Elemente vom Rang 3 und vom Rang 7 in A an.

c)

> Bestimmen Sie die korrekte Anzahl der Vergleiche, die MERGE-SORT auf der Input-Folge $A = [8, 1, 3, 5]$ ausführt.

d)

> Was ist Input und Output von SELECTION? Was ist die algorithmische Aufgabe?

e)

> $A = [a_1, \ldots, a_n], B = [b_1, \ldots, b_n]$ aufsteigend sortierte Felder von $2n$ paarw. versch. nat. Zahlen. Algorithmus angeben, der in $\mathcal{O}(\log n)$ Schritten den Median m des Feldes $X = [a_1, \ldots, a_n, b_1, \ldots, b_n]$ findet.

8. Aufgabe

> Die Datenstruktur SCHLANGE arbeitet nach dem FIFO-Prinzip (First In First Out). Es gibt die Operationen ENQUEUE und DEQUEUE. ENQUEUE fügt ein Element an das Ende der Schlange ein. DEQUEUE entfernt das Element am Kopf der Schlange.

a)

> Geben Sie mit Hilfe von zwei Stapeln eine Implementierung der Datenstruktur SCHLANGE mit den beidne Operationen ENQUEUE und DEQUEUE an. Beide Operationen sollen $\mathcal{O}(1)$ amortisierte Zeit benötigen.

b)

> Zeigen Sie mit Hilfe der Potentialmethode, dass bei Ihrer Implementiertung beide Operationen $\mathcal{O}(1)$ amortisierte Zeit benötigen.

9. Aufgabe

> Ein ungerichteter Graph $G = (V, E)$ heißt bipartit, wenn es disjunkte Knotenmengen $A, B \subset V$ mit $A \cup B = V$ gibt, sodass jede Kante $e \in E$ einen Endpunkt in A und einen in B hat. Beschreiben Sie einen möglichst effizienten Algorithmus, der für einen ungerichteten Graphen testet, ob er bipartit ist. Analysieren Sie seine Laufzeit.

> Es kommt nicht darauf an, den Algorithmus in Pseudocode aufzuschreiben. Wichtiger ist, die Hauptschritte Ihres Algorihtmus kurz und klar verbal zu beschreiben und jeweils eine Aussage über die Laufzeit zu treffen.

10. Aufgabe

a)

> Beschreiben Sie den Algorithmus HEAPSORT. Erläutern Sie die einzelnen Schritte und die jeweilige Laufzeiten.

b)

> Verringert sich die asymptotische Laufzeit, wenn HEAPSORT auf ein bereits aufsteigend sortiertes Feld angewendet wird?

c)

> Führen Sie HEAPSORT auf dem Array $A = [7, 10, 18, 12, 31, 4, 22, 26]$ aus.

d)

> Gegeben sei ein binärer Heap, der n Schlüssel verwaltet. Zeigen Sie, dass er mindestens $\Omega(n \log n)$ Zeit erfordert, eine aufsteigende sortierte Folge dieser Schlüssel zu erstellen.

11. Aufgabe

> Gesucht ist eine Datenstruktur zur Speicherung von n Schlüsseln $x_i \in \mathbb{R}$, die INSERT und DELETE in Zeit $\mathcal{O}(\log n)$ erlaubt udn folgende Operationen unterstützt:
>
> - GREATERTHAN(a) zur Ausgabe aller Schlüssel x_i, die in der Datenstruktur gespeichert sind und für die $a \in x_i$ gilt.
> - BETWEEN(a,b) zu Ausgabe aller Schlüssel x_i, die in der Datenstruktur gespeichert sind und für die $a \leq x_i$ gilt.

a)

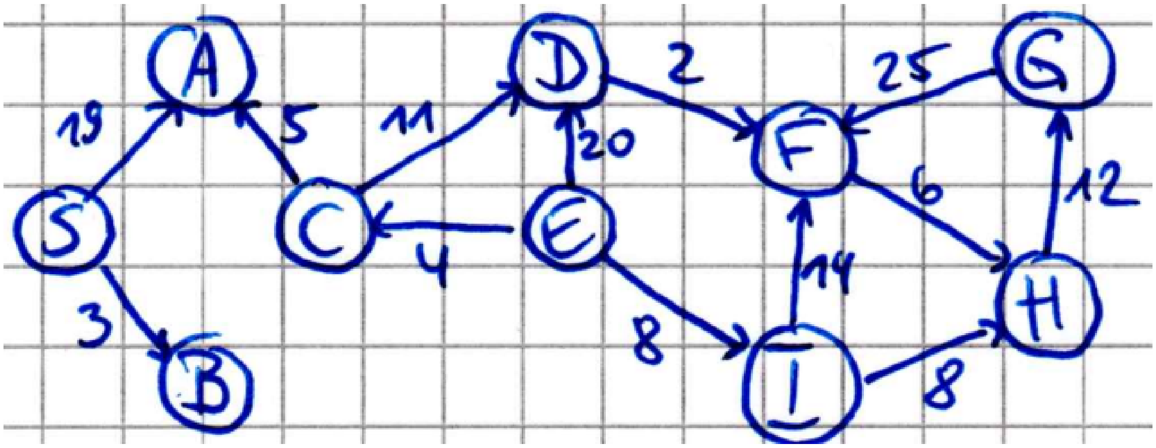
> Beschreiben Sie eine Implementierung von GREATERTHAN, die es erlaubt, die Suchanfrage für Eingabe a in $\mathcal{O}(m + \log n)$ Zeit zu beantworten. dabei sind n die Anzahl der in der Datenstruktur gespeicherten Schlüssel und m die Anzahl der ausgegebenen Schlüssel.

b)

> Wie kann die Operation BETWEEN(a,b) in $\mathcal{O}(m + \log n)$ Zeit (n,m wie oben) ausgeführt werden?

Es kommt nicht darauf an, den Algorithmus in Pseudocode aufzuschreiben. Wichtiger ist, die Hauptschritte Ihres Algorithmus kurz und klar verbal zu beschreiben und jeweils eine Aussage über die Laufzeit zu treffen.

12. Aufgabe



a)

$G = (V, E)$ gewichteter Graph. $w : E \to \mathbb{N}$ Kantengewichtsfkt. $S \in V$. Definiere Shortest Path Tree von G mit Wurzel S.

b)

Shortest Path Tree kann mit Dijkstra berechnet werden. Welche wesentliche Datenstruktur verwendet der Algorithmus?

c)

Dijkstra anwenden. Nach jedem Knoten die Elemente mit ihrem Schlüssel angeben.