Self-Study Session 02

Algorithms and Data Structures (DAT2, SW2, DV2)

Instructions. You have to **work individually** on these questions from 8:15 to 10:00. From 10:00 to 12:00 you can join your group and discuss your solutions togther.

- Read carefully the text of each exercise before solving it! Pay particular attentions to the terms in bold.
- If a question seems ambiguous, write under which assumptions you are solving it.
- You can use **CLRS** to refer to the textbook T.H. Cormen, Ch. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms* (3rd edition) in your answers.
- Make an effort to use a readable handwriting and to present your solutions neatly.
- The TAs will be available from 10:00 to 12:00. Use the digital trashcan to ask for help.
- The points relative to each question give an indication of their complexity relative to this exercise sheet. These points **may not** correspond to the amount of points you may get for a similar exercise at the exam.

Question 1. 20 Pts

Identifying asymptotic notation. (Note: lg means logarithm in base 2)

(1.1) [5 Pts] Mark **ALL** the correct answers. $n \lg n^5 + n \lg 2^n + n\sqrt{n}$ is

☐ **b)** $\Theta(n \lg n)$ ☐ **b)** $\Theta(n)$ ☐ **c)** $\Theta(\sqrt{n})$ ☐ **d)** $\Theta(n^2)$ ☐ **e)** $\Theta(n^{1.5})$

(1.2) [5 Pts] Mark **ALL** the correct answers. $n \lg n^5 + n \lg 2^n + n\sqrt{n}$ is

☐ **a)** $\Omega(\lg n)$ ☐ **b)** $O(n)$ ☐ **c)** $\Omega(\sqrt{n})$ ☐ **d)** $O(n^2)$ ☐ **e)** $O(\sqrt{n})$

(1.3) [5 Pts] Mark **ALL** the correct answers. $700 \cdot n^2 + \frac{n^2 \lg n}{999} + \lg n^n$ is:

☐ **a)** $\Omega(n \lg n)$ ☐ **b)** $O(n^3)$ ☐ **c)** $O(n^2)$ ☐ **d)** $\Omega(n^2 \lg n)$ ☐ **e)** $O(n^2 \lg n)$

(1.4) [5 Pts] Mark **ALL** all the functions below that satisfy $\lg(f(n) \cdot g(n)) = \Theta(n)$.

☐ **a)** $f(n) = n^2$, $g(n) = n^2$ ☐ **b)** $f(n) = 2^n$, $g(n) = 2^n$

☐ **c)** $f(n) = 2^n$, $g(n) = 4^n$ ☐ **d)** $f(n) = n \lg n$, $g(n) = 2^n$

Question 2. 20 Pts

Consider the following recurrences

$$T(n) = \begin{cases} \Theta(1) & \text{if } n \in \{0, 1\} \\ T(n-1) + T(n-2) + \Theta(1) & \text{if } n > 1 \end{cases} \qquad Q(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ 8 \cdot Q(n/2) + n^4 & \text{if } n > 1 \end{cases}$$

(2.1) [5 Pts] Mark **ALL** correct answers.

- ☐ **a)** $T(n)$ can be solved using Case 1 of the Master Theorem
- ☐ **b)** $Q(n)$ can be solved using Case 2 of the Master Theorem
- ☐ **c)** $Q(n)$ can be solved using Case 3 of the Master Theorem
- ☐ **d)** $T(n)$ cannot be solved using the Master Theorem

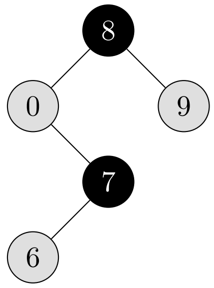
(2.2) [5 Pts] Mark **ALL** correct answers.

- ☐ **a)** $Q(n) = \Theta(n^2 \lg n)$
- ☐ **b)** $Q(n) = \Omega(n)$
- ☐ **c)** $Q(n) = \Theta(n^4)$
- ☐ **d)** $Q(n) = O(n^3)$

(2.3) [10 Pts] Prove that $T(n) = O(2^n)$ using the substitution method.

Question 3. 20 Pts

(3.1) [4 Pts] Mark **ALL** the correct statements. Consider the tree T depicted below



- ☐ **a)** The height of T is 3
- ☐ **b)** T satisfies the binary-search tree property
- ☐ **c)** T satisfies the red-black tree property (NIL nodes are omitted)
- ☐ **d)** T satisfies the max-heap property
- ☐ **e)** T corresponds to the array $[8, 0, 9, 7, 6]$ interpreted as a binary tree.

(3.2) [6 Pts] Consider the hash table $H = \text{NIL}, \text{NIL}, 46, 35, \text{NIL}, 15, 92, \text{NIL}, 52, \text{NIL}, 87$. Insert the keys $44, 84, 17, 20$ in H using *open addressing* with the auxiliary function $h'(k) = k$.

Mark the hash table resulting by the insertion of these keys using linear probing.

- ☐ **a)** $44, 17, 46, 35, \text{NIL}, 15, 92, 84, 52, 20, 87$
- ☐ **b)** $44, \text{NIL}, 46, 35, 17, 15, 92, 84, 52, 20, 87$
- ☐ **c)** $44, 20, 46, 35, \text{NIL}, 15, 92, 84, 52, 17, 87$
- ☐ **d)** none of the above

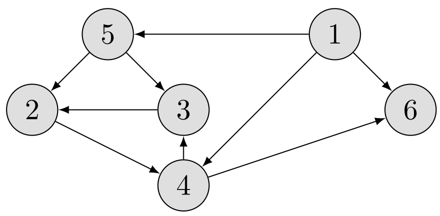
Mark the hash table resulting by the insertion of these keys using quadratic probing with $c_1 = 2$ and $c_2 = 4$.

- ☐ **a)** $44, 17, 46, 35, \text{NIL}, 15, 92, 84, 52, 20, 87$
- ☐ **b)** $44, \text{NIL}, 46, 35, 17, 15, 92, 84, 52, 20, 87$
- ☐ **c)** $44, 20, 46, 35, \text{NIL}, 15, 92, 84, 52, 17, 87$
- ☐ **d)** none of the above

Mark the hash table resulting by the insertion of these keys using double hashing with $h_1(k) = k$ and $h_2(k) = 1 + (k \mod (m-1))$.

- ☐ **a)** $44, 17, 46, 35, \text{NIL}, 15, 92, 84, 52, 20, 87$
- ☐ **b)** $44, \text{NIL}, 46, 35, 17, 15, 92, 84, 52, 20, 87$
- ☐ **c)** $44, 20, 46, 35, \text{NIL}, 15, 92, 84, 52, 17, 87$
- ☐ **d)** none of the above

(3.3) [10 Pts] Consider the directed graph G depicted below.



(a) Write the intervals for the discovery time and finishing time of each vertex in the graph obtained by performing a depth-first search visit of G. *Remark:* If more than one vertex can be chosen, choose the one with smallest label.

(b) Write the corresponding "parenthesization" of the vertices in the sense of Theorem 22.7 in CLRS

(c) Assign to each edge a label T (tree edge), B (back edge), F (forward edge), or C (cross edge) corresponding to the classification of edges induced by the DFS visit performed before.

(d) Show the components computed by STRONGLY-CONNECTED-COMPONENTS(G).

Question 4. 20 Pts

A student participates in an exam which includes n different questions $Q_1, \ldots, Q_n$. Each question Q_i ($i = 1 \,..\, n$) has a positive value in points, denoted by p_i, and also the time (in minutes) to solve the question, denoted by t_i. Overall, the exam concludes after T minutes. Given this information, the student wants to find a selection $S \subseteq \{1, \ldots, n\}$ of questions to solve that maximises the total amount of points $\sum_{i \in S} p_i$ subject to the time constraint $\sum_{i \in S} t_i \leq T$.

(a) [5 Pts] Argue why brute-force enumeration of all possible selections of questions leads to an exponential algorithm?

(b) [10 Pts] Describe a bottom-up dynamic programming procedure PointsQuestions(p, t, T) that takes as input $p[1 \,..\, n]$, $t[1 \,..\, n]$, and $T > 0$ and returns the the total amount of points of an optimal selection of questions. Analyse the worst-case running time of your algorithm.

(c) [5 Pts] Describe a procedure Questions(p, t, T) that prints an optimal selection of questions.

Question 5. 20 Pts

Let A denote a finite set of labels. We extend the notion of directed graphs to labelled directed graphs $G = (V, E)$ were V is a set of vertices and $E \subseteq V \times A \times V$ is a set of labelled edges. Each labelled edge $(v, a, v') \in E$ is associated to a weight $w(v, a, v')$ by means of a weight function $w \colon E \to \mathbb{R}$.
Let $\pi = \langle v_0, a_1, v_1, a_1, \dots, a_k v_k \rangle$ be a path of G, the *weight* of p is the sum of the weights of its constituent edges, namely $w(\pi) = \sum_{i=1}^{k} w(v_{i-1}, a_i, v_i)$; and the *trace* of p is the sequence of labels of p, namely $tr(\pi) = a_1 \mathinner{.\,.} a_k$.
All-pairs same-trace shortest paths problem: Given a labelled directed graphs $G = (V, E)$ and a trace $\tau = a_1 \mathinner{.\,.} a_k$, we want to find for every pair of vertices $u, v \in V$ a shortest (least-weighted) path $u \leadsto_\pi v$ from u to v such that $tr(\pi) = \tau$.

Hint: By assuming $V = \{1, \dots, n\}$ and $A = \{1, \dots, m\}$, we can conveniently represent a labelled directed graph $G = (V, E)$ with weight function w by means of a sequence $W^1 \mathinner{.\,.} W^m$ of $n \times n$ adjacency matrices. For $a \in A$ the entry for the pair $(i, j) \in V \times V$ of $W^a = (w^a_{ij})$ is

$$w^a_{ij} = \begin{cases} w(i, a, j) & \text{if } (i, a, j) \in E \\ \infty & \text{if } (i, a, j) \notin E \end{cases}$$

(a) [10 Pts] Describe an algorithm ALL-PAIRS-SAME-SEQUENCE(W, τ) that, given an array $W[1 \mathinner{.\,.} m]$ of $n \times n$ adjacency matrices and an array $\tau[1 \mathinner{.\,.} k]$ of labels, returns an $n \times n$ matrix $D = (d_{i,j})$ such that $d_{i,j} = \min\{w(\pi) \mid i \leadsto_\pi j, \text{ and } tr(\pi) = \tau\}$.

(b) [10 Pts] Analyse the asymptotic worst-case running time and space usage of your solution.