DELFT UNIVERSITY OF TECHNOLOGY
Faculty of Electrical Engineering, Mathematics and Computer Science

Exam TI2306 Algorithm Design

February 1, 2017, 13.30-16.30

- Usage of the book, notes or a calculator during this test is not allowed.
- This exam contains 7 open questions (worth a total 65 points) contributing 6.5/10 to your grade, and 15 multiple choice questions contributing 3.5/10 towards your grade. Note that the final mark for this course also consists for $\frac{1}{3}$ of the unrounded score for the lab course (if both ≥ 5.0).
- If your average grade for the homework exercises of this run of the course is at least 5.8, the grade for this exam is increased by 0.5 (but never beyond 10.0).
- The open questions require about two hours of your time:
  - Please answer in clear English or Dutch and write legible (first use pencil or scrap paper).
  - Do not give any irrelevant information. This might lead to subtraction of points. The indication of the number of lines is appropriate for typed lines (extra space caused by your handwriting and use of white space will not be held against you).
  - If an algorithm is asked, please provide the most efficient algorithm you can think of. Providing a suboptimal algorithm can lead to a subtraction of points.
  - If asked for pseudocode, you may call algorithms from the book unless specifically asked for the respective pseudocode.
  - Before handing in your solutions, please verify that your name and student number are on every page. Provide the course code and the number of pages (at least) on the first page.
- Regarding the multiple choice questions:
  - Each question has only one correct answer.
  - All questions count equally towards the grade. In computing the grade we will correct for the effects of random guessing. If you do not provide an answer, this is treated as a wrong answer.
  - Write your answers on scrap paper before copying them to the answer form.
  - Fill in your student number using digits as well as using the blocks.
  - Preferably use a dark pen or pencil for filling in the form and limit the number of corrections.
  - Try to spend no more than 1 hour on the multiple choice questions.
- The exam material consists of Chapters 4–7 from Algorithm Design by Kleinberg and Tardos (except sections 4.9*, 5.6, 6.10*, 7.4*, en 7.13*), as well as the notes on the Master Method and Proving techniques (guide), also applied to simple graph algorithms (from Chapter 3).
- Total number of pages (without this preamble): 5.

Open questions

1. (10 points) Given is a directed graph $G = (V, E)$. Prove that if there are no cycles in G, then there is a vertex without incoming edges (in 10 lines).

   Recall the following *hints for a good proof*: 1. Indicate the proving techniques you are using. 2. If necessary, define your notation. 3. Make assumptions explicit. 4. Refer to propositions you are using.

2. Suppose you have n jobs and one machine. Job i requires t_i time on the machine. The machine can only execute one job at a time. The goal is to determine the completion time c_i for each job i such that their sum ($\sum_i c_i$) is as small as possible. For example: running job $t_1 = 4$ first and then $t_2 = 6$, results in completion times $c_1 = 4$ and $c_2 = 10$ and thus a sum of $4 + 10 = 14$.

   (a) (6 points) Give pseudocode of a greedy algorithm that determines for a series of n jobs the schedule for the machine minimizing the sum of completion times. Compute these completion times $c_1, c_2, \ldots, c_n$ as part of the algorithm. Briefly explain your pseudocode.

   (b) (2 points) Give a tight upper bound on the run time of your algorithm and explain.

3. Given is a set S of points in a plane. The (known) algorithm for finding the closest pair of points contains two recursive calls: one on the points left and one on the points right of a line in the middle, $x = L$. The smallest distance between the two closest pairs thus found we denote by δ.

   (a) (3 points) How can you determine in linear time whether δ is the overall smallest distance between any two points in S?

   (b) (6 points) Prove why your answer to (a) is correct.

4. (6 points) Give an asymptotic tight upper and lower bound for $T(n)$ in the following recurrence relation. Indicate the steps you use to get to your solution (in 5 lines).

   $T(n) = 3T(n/3) + n$

5. Delft University of Technology likes to place large screens along the A4 to advertise research. Rijkswaterstaat has indicated n possible locations for these screens. These locations are identified by their distance from Amsterdam, in increasing order: $x_1, x_2, \ldots, x_n$. A preliminary study derived for each of the locations an expected (positive) effect $v_1, v_2, \ldots, v_n$. Also it is assumed that the total effect is equal to the sum of these expectations for the selected locations. Rijkswaterstaat requires a minimum distance of 5km between the screens.

   (a) (6 points) Give a recursive function to express the maximum possible total effect. Indicate which argument(s) the first call to this function requires, and briefly explain your formula.

   (b) (5 points) Give pseudocode of an efficient, *iterative* algorithm to compute the maximum possible total effect. (Indicate this result, e.g., by `return` or `print`.)

6. The Norwegian Railways (NR) have to find out whether they have enough trains. The Norwegian rail network is divided into n trajectories which all take exactly one day to cover. For each trajectory $i \in \{1, 2, \ldots, n\}$ the Norwegian government has set a minimum (c_i^-) and a maximum (c_i^+) number of trains per day. A complicating factor is that for each trajectory i only certain train types j can be used. For every type of train $j \in \{1, 2, \ldots, k\}$ the number of trains of this type the NR has available (t_j) is given. The question is whether this is sufficient to meet all criteria.

   (a) (5 points) Model this problem as a circulation with lower bounds. Define all aspects of your model (in 8 lines). Also draw a simple example with three trajectories and two types of trains.

   (b) (4 points) Explain how to determine with this circulation whether the NR has enough trains (in 6 lines).

   (c) (2 points) Explain how to obtain a feasible allocation of trains to trajectories.

7. (10 points) Given a flow network and a valid flow f on this network. Prove the following statement.

   If no augmenting path of f exists, then there is an s-t cut (A, B) such that $v(f) = \text{cap}(A, B)$.
   (Hint: use the *flow value lemma*.)

Meerkeuzevragen

1. Given $f(n) = n^2$ and $g(n) = n^2 \log n$. Which statement is True?
   A. $f(n)$ is $O(g(n))$.
   B. $g(n)$ is $\Theta(f(n))$.
   C. $f(n)$ is $\Omega(g(n))$.
   D. $g(n)$ is $O(f(n))$.

2. Let an undirected graph G with n nodes be given. Which of the following statements is ***not*** True?
   A. If G has at least n edges, then G is connected.
   B. If G has at least n edges, G contains a cycle.
   C. If G is a tree, then G has at most $n-1$ edges.
   D. If G is connected, then G has at least $n-1$ edges.

3. Given a set of n open intervals (start time, end time) in $\mathbb{R}^+$. How would you sort these intervals to obtain in $\mathcal{O}(n \log n)$ time...

   1. ...the largest number of overlapping intervals (interval partitioning)? and
   2. ...the largest number of non-overlapping intervals (interval scheduling)?

   Sort
   A. 1. decreasing on start time and 2. decreasing on start time
   B. 1. increasing on start time and 2. increasing on end time
   C. 1. decreasing on end time and 2. increasing on start time
   D. 1. increasing on end time and 2. increasing on start time

4. For a text, the frequency of each character has been given. Moreover, let the optimal prefix code for this text be given as a binary tree. Which statement is correct?
   A. The two least frequent characters have the same parent (are siblings).
   B. The two least frequent characters are at the deepest level in the tree.
   C. The tree is balanced.
   D. Each of these statements is correct.

5. Give an asymptotic tight upper and lower bound for $T(n)$ in the following recurrence relation:
   $T(n) = 6T(n/2) + n^3$
   A. $\Theta(n^2 \log n)$
   B. $\Theta(n^3)$
   C. $\Theta(n^3 \log n)$
   D. $\Theta(n^3 \log^2 n)$

6. Two very large, sorted arrays with numbers are given, and both have length $\Theta(n)$. What is a tight asymptotic bound on the run time of an algorithm that determines the median of the combined set of numbers as efficiently as possible?[1]
   A. $\Theta(\log n)$
   B. $\Theta(n)$
   C. $\Theta(n \log n)$
   D. $\Theta(n^2)$

[1] The median of a set is equivalent to the middle element of a sorted array of all elements.

7. Karatsuba has proposed an efficient recursive (divide-and-conquer) algorithm to multiply two very large integers. Which recurrence relation correctly describes the run time of this algorithm?

   A. $T(n) = 2T(n/2) + cn$

   B. $T(n) = 3T(n/2) + cn$

   C. $T(n) = 2T(n/3) + cn$

   D. $T(n) = 3T(n/3) + cn$

8. Given is a set of points $(x_1, y_1), \ldots, (x_n, y_n)$, sorted on their x coordinate. Also given is for each pair $i < j$ the (minimal) error $e_{i,j}$ obtained in case points i to j are approximated by a single line segment.[2] Additionally, costs C are given for the introduction of an additional line segment.

   In the following algorithm to compute the minimal costs for approximating points 1 tot n with line segments, one line is incomplete.

```
1  for j ← 1 to n do
2      M[j] ← e_{1,j} + C
3      for i ← 2 to j // als j ≥ 2
4      do
5          m ← ...
6          if m < M[j] then
7              M[j] ← m
8          end
9      end
10 end
11 print M[n]
```

   Which of the following completions of line 5 achieves the desired result?

   A. $m \leftarrow e_{i,j} + C + M[i-1]$

   B. $m \leftarrow e_{1,j} + C + M[j]$

   C. $m \leftarrow \min\{e_{i,j} + C, M[i-1]\}$

   D. $m \leftarrow \min\{e_{i,j}, C + M[i]\}$

9. Given are n jobs and for each of the jobs $i \in \{1, \ldots, n\}$ a fixed (integer) duration w_i. We have W time available on a machine to execute these jobs. The objective is to select a subset S of the jobs such that $\sum_{i \in S} w_i \leq W$ and such that $\sum_{i \in S} w_i$ is as large as possible. What is a tight worst-case upper bound on the run time of the most efficient implementation for solving this problem?

   A. $\mathcal{O}(n)$

   B. $\mathcal{O}(n \log n)$

   C. $\mathcal{O}(n^2)$

   D. $\mathcal{O}(nW)$

[2] The error is defined as the sum of the squared Euclidean distances from the line segment to the points i to j.

10. At your birthday you received P euro instead of the presents on your wish list. For each of the n presents on your list you have found its price p_i. You decide to use as much of the received money as possible to buy the presents on your list. This problem can be solved efficiently using dynamic programming and the following recursive function.

$$\mathrm{OPT}(n, P) = \begin{cases} 0 & \text{if } i = 0, \\ \mathrm{OPT}(i-1, p) & \text{if } p_i > p \\ \ldots & \text{otherwise} \end{cases}$$

What needs to be on the dots (...) to make sure that this function describes the maximal amount you could spend on the presents on your list?

A. $\max_{1 \leq j \leq i}\{\mathrm{OPT}(j, p - p_j) + p_j\}$
B. $\max\{\max_{1 \leq j \leq i}[\mathrm{OPT}(j, p - p_j) + p_j], \mathrm{OPT}(i-1, p)\}$
C. $\max_{1 \leq j \leq i}\{\mathrm{OPT}(i-j, p - p_j) + \mathrm{OPT}(j, p_j) + p_j\}$
D. $\max\{\mathrm{OPT}(i-1, p), \mathrm{OPT}(i-1, p - p_i) + p_i\}$

11. The problem of understanding the two-dimensional structure of a RNA-molecule (RNA Secondary Structure) is defined as follows: given a sequence of bases $B = b_1, \ldots, b_n$ with $b_i \in \{A, U, C, G\}$, find a set S of base pairs (i, j) that is as large as possible and meets a certain set of criteria. If a base pair (i, j) meets these criteria, we write $p(i, j)$. The following function then describes the size of the largest set of base pairs that meet these criteria.

$$\mathrm{OPT}(i, j) = \max\left\{\mathrm{OPT}(i, j-1), \max_{\{t \mid i \leq t < j-4 \wedge p(t,j)\}} \{1 + \mathrm{OPT}(i, t-1) + \mathrm{OPT}(t+1, j-1)\}\right\}$$

In which order can the subproblems be computed in a dynamic program?

A.
```
for i := 1 to n − 1 do
    for j := i to n do
        compute OPT(i, j)
    end
end
```

B.
```
for i := n − 1 downto 1 do
    for j := i downto 1 do
        compute OPT(i, j)
    end
end
```

C.
```
for k := 1 to n − 1 do
    for i := 1 to n − k do
        compute OPT(i, i + k)
    end
end
```

D.
```
for i := 1 to n − 1 do
    for k := 1 to n − i do
        compute OPT(i, i + k)
    end
end
```

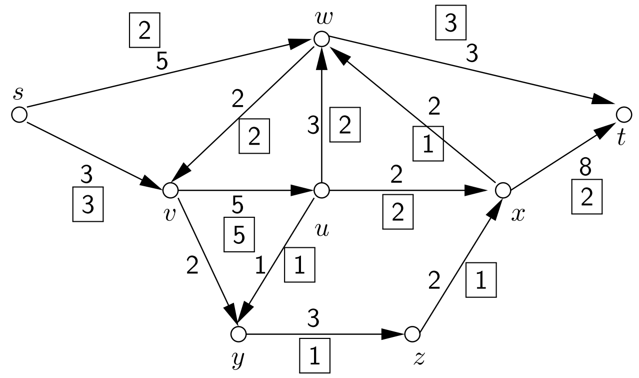

Figure 1: A flow network. The numbers next to the edges are the capacities. The flow on an edge is indicated in a square.

12. Given is a directed graph $G = (V, E)$ with $V = \{u, v, w\}$ and the following (directed) edges with lower and upper bounds.

edge	lower bound	upper bound
(u, v)	4	7
(v, w)	5	9
(w, u)	2	10

Consider a translation of this problem to a network flow problem G' without lower bounds and without demands, such that a maximum flow of G' indicates whether a valid circulation in G exists. Does an edge (s, v) or (v, t) exist in G' and what is the capacity c of this edge?

A. $c(v, t) = 1$

B. $c(v, t) = 2$

C. $c(s, v) = 1$

D. $c(s, v) = 3$

13. Consider the flow network in Figure 1. With what amount can the s-t flow be augmented? (If it helps you, you could draw the residual graph.)

A. 0

B. 1

C. 2

D. 3

14. Again, consider the flow network from Figure 1. What is the capacity of the cut $(\{s, w, v\}, \{u, y, z, x, t\})$?

A. 2

B. 5

C. 8

D. 10

15. A variant of the algorithm by Ford-Fulkerson exists that is more efficient due to a so-called scaling technique. Which of the following describes best the idea of this technique?

A. In every iteration, find a flow of at most Δ and halve Δ after every iteration.

B. In every iteration, find a flow of at least Δ and double Δ after every iteration.

C. In every iteration, consider only edges with a residual capacity of at least Δ and halve Δ after every iteration.

D. In every iteration, consider only edges with a residual capacity of at least Δ and double Δ after every iteration.

End of exam questions