Final exam Algorithmics SS 2023

Prof. Dr. Sebastian Iwanowski **30.08.2023**

Time limit: 120 minutes

Admitted appliances: none (calculator allowed if you really need this)

Please give your answers and interim results exclusively in the pages of the assignments. If the space is not sufficient, you may use the blank reverse sheet on the opposite side. The same holds for scratch paper. Please indicate clearly which parts should be graded and which not.

Language: You may answer each assignment in German or English just as you feel most comfortable in order to express your thoughts and intentions clearly. In particular, you may also switch the language between or within the assignments.

This exam issues 50 + 3 bonus evaluation credits (EC).
For passing this exam you need at least 25 EC.

Good luck!

Assignment 1: 2 EC

Sort the following complexity classes by inclusion:

a) $O(n^2 (\log_2 n)^3)$ b) $O(n^2 (\log_3 n)^2)$ c) $O(n!)$ d) $O(n^{2,01})$

e) $O(n (\log_3 n)^3)$ f) $O((\log_{1.01} n)^{1000})$ g) $O(n^2)$

Show a linear chain of set inclusion and indicate which inclusion is proper and which is an equality.

For simplicity, you may just work with the letters a) to g).

Assignment 2: 3 EC

Explain the two non-equivalent definitions of Theta:

Name one algorithm as an example where the Theta definition for a certain run time holds in one definition but does not hold in the other.
State the actual run time and justify your answer.

Assignment 3: 4 EC

Analyse Mergesort:

a) Develop a recursive formula for T(n) which should be the worst-case run time of Mergesort with n items. (1 EC)

b) Give the tightest worst-case complexity class of T(n) and prove this by mathematical induction using a) (3 EC)

For simplicity, you may assume that n is a power of 2.

Assignment 4: 2 EC

a) Which is the assumption for a sorting problem in order to claim that it has got a lower bound of Ω (n log n) in worst case?

b) Which algorithm violates this assumption and has got a better run time in worst case?
Tell which is the assumption for this algorithm and denote its complexity class. You need not give any details of this algorithm.

Assignment 5: 4 EC

a) Specify which problem has to be solved with the interface Select (k,A) by explaining the parameters. (1 EC)

b) Describe a trivial algorithm for solving a) in words and tell its asymptotic run time in the worst case (without proof). (2 EC)

c) For the deterministic algorithm solving a) in linear time, specify exactly the parameter of the above interface on which the run time behaviour depends linearly and specify the other parameter on which the run time behaviour does not depend at all. (1 EC)

Assignment 6: 4 EC

a) Simulate Radixsort with Countingsort using the following example:

   miss, messy, class, kiss

   Start with the order given here and give all the orders obtained in between until the words are sorted. (2 EC)

b) What is the worst-case run time of the general algorithm simulated in a) for the particular example? Explain the meaning of all parameters. (2 EC)

Assignment 7: 2 EC

For the dictionary problem, tell in which application scenario hashing is advantageous to balanced trees and for which not.

Assignment 8: 5 EC

a) Define the data structure of an a-b-tree: For this, specify all conditions for a,b as well as the number of children. (2 EC)

b) What is the asymptotic run time of the 3 operations of the dictionary problem in an a-b-tree? Name these operations and state the dependence on n, where n is the number of data currently stored. (1 EC)

c) Which a-b-tree can be transformed into a red-black tree by local modification of the nodes?
Show the transformation of this special a-b tree to red-black in a sketch: Distinguish all different cases to be considered. (2 EC)

Assignment 9: 5 EC

a) Which is the exact task of an abstract union-find algorithm defined for arbitrary sets? Specify the two functions provided. (1 EC)

b) Explain the efficient solution for this problem for the example {{1, 2, 3}, {4, 5}, {6, 7, 8, 9}} in the following way:
Sketch an appropriate optimum **graphical** structure storing the data. Then show, how the first and the second set are united. Finally, show how the result is united with the third set.(2 EC)

c) Which is the run time for the operations defined in a) in general? (1 EC)

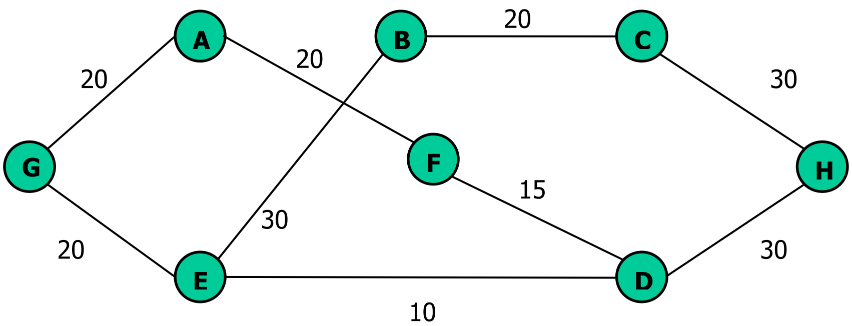
d) For which application is a union-find algorithm essential in order to get an efficient solution? Explain exactly what are the sets in this application. (1 EC)

Assignment 10: 3 EC

a) What is the run time of the dynamic programming algorithm of Floyd-Warshall for the All Pairs Shortest Path (APSP) problem? Denote the complexity class for n nodes.(1 EC)

b) Specify the condition when Floyd-Warshall may be advantageous compared to an iterative straight forward solution with Dijkstra's algorithm (no Fibonacci Heaps) when each node is taken as a source subsequently (give the run time of APSP using Dijkstra).
Define a condition often met in practice when iterative Dijkstra is definitely better (give also the run time for that case). (2 EC)

Assignment 11: 3 EC



a) Determine the maximum matching in the above graph using the following method:
Start with the matching (A.F); (G,E); (C,H) and increase this according to Edmonds' algorithm. (2 EC)

b) Why is it not possible to find the maximum matching using the maximum flow algorithm of Edmonds-Karp here? Justify your answer. (1 BE)

Assignment 12: 6 EC

a) State the 3 equivalent conditions of Ford-Fulkerson for optimum flow algorithms. (2 EC)

b) Prove one of the 3 implications required for the full proof of equivalence. (2 EC)

c) 2 of the above conditions are exploited directly by the Edmonds-Karp algorithm. Why is the equivalence of these 2 conditions required and not just one direction of the implication? (2 EC)

Assignment 13: 3 EC

a) Illustrate the prefix funtion by giving the result for the following pattern:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
a	a	a	a	b	b	b	a	a	a	b	b	b	a	a	a	a	a	b	b	b	a	a	b

b) Argue why the KMP procedure has a run time of O(n):

```
i := 1; q := 0;
while i ≤ n do {
   while (q>0) and (T[i] ≠ P[q+1]) do
      q := π (q);
   if T[i] = P[q+1] then q := q+1;
   if q = m
      then {
         print („Matching at position “, i-m);
         q := π (q);
      }
   i := i+1;
}
```

Assignment 14: 4 EC

Consider the problem minimum spanning tree in the plane:

a) Sketch a naive algorithm and denote and justify its complexity class. (2 EC)

b) Sketch how this problem can be solved more efficiently using a Voronoi diagram. Denote and justify the complexity class also considering the effort for computing the Voronoi diagram.
Hint: The run time using the Voronoi diagram is not trivial but depends on a property of the Voronoi diagram which should be mentioned here. (2 EC)

Assignment 15:

Bonus 3 EC

Consider Fortune's plane sweep algorithm for the construction of a Voronoi diagram:

Consider a spike event where the reference points belonging to the two intersecting spikes are denoted by $P_1=(x_1,y_1)$, $P_2=(x_2,y_2)$, $P_3=(x_3,y_3)$ and where the y coordinates of the parabolic segments belonging to these 3 reference points are in descending order.
Denote the intersection point of the spikes by (x_0,y_0).
Let $P_0=(x_0,y_0)$ be the other reference point of the next spike above and $P_4=(x_4,y_4)$ be the other reference point of the next spike below.

Answer the following questions:

a) Which reference point belongs to the parabola vanishing at the spike event?

b) Which new spike has to be computed, and which spike events may eventually be inserted into the EPS?