Final exam Algorithmics SS 2017

Iwanowski 09.08.2017

Hints:

Time limit: 120 minutes

Admitted appliances: none

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.

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 consists of 13 pages including this cover sheet.

This exam issues 50 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) $O(n^3 (\log_2 n)^2)$

b) $O(n^2 (\log_3 n)^3)$

c) $O(n!)$

d) $O(n^{2,01})$

e) $O(n^2 (\log_2 n)^3)$

f) $O(n^3 (\log_3 n))$

g) $O(n^2)$

Show a linear chain 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: (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 3: (2 EC)

a) Which is the assumption that each sorting problem 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 4: (8 EC)

a) Show the first steps of the optimum select algorithm Select (5,A) for the following input:

A = [19, 7, 1, 4, 18, 6, 24, 15, 9, 8, 3, 25, 11, 20, 5, 23, 17, 10, 14, 12, 16, 22, 13, 21, 2]

Give all intermediate results of one procedure call comprising of all intermediate steps: You should finish with the last recursive call of the first invocation. For that recursive call you should give the parameters but you need not continue and you need not give the final result. For the first recursive call in this procedure (which occurs earlier in this procedure) you need only give the result and continue with the next step using this result. (5 EC)

b) Give a recursive equation for the upper bound of the run time and denote the complexity class (no proof required). (2 EC)

c) In order to achieve the complexity class of b), what is the crucial parameter in the algorithm shown in a)? Explain which parameter value is eligible and which not in order to achieve the desired complexity class. (1 EC)

Assignment 5: (2 EC)

a) What is the improvement of quadratic binary search compared with ordinary binary search? Argue with complexity classes.

b) Is there also a deterioration? Argue again with complexity classes.

Assignment 6: (2 EC)

How does Hashing compare with balanced trees if you want to maintain a dictionary with n data? Denote the differences in time and space and clarify if your statement relates to average or worst-case behaviour.
Hint: Hashing depends on a second parameter. Explain also this parameter.

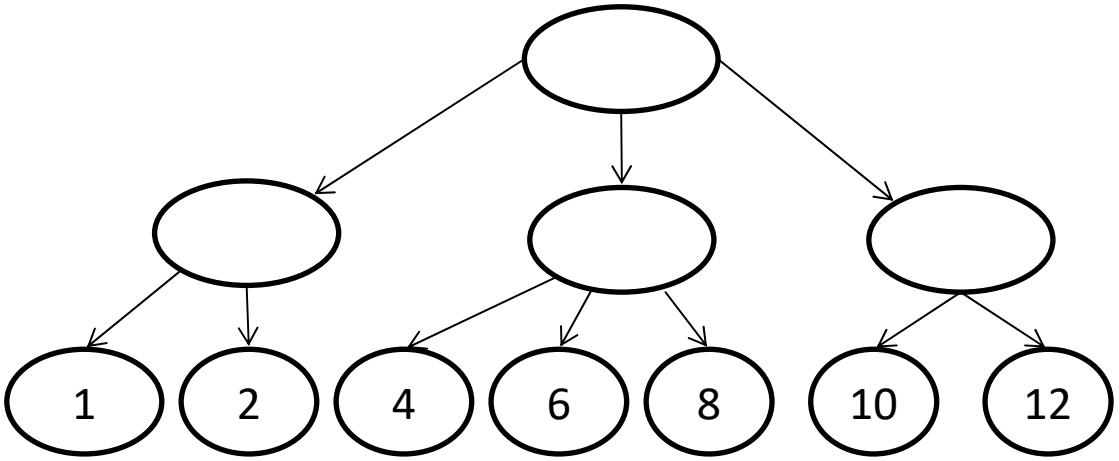
Assignment 7: (5 EC)

In the (2,3) tree below identify the data with the keys.

a) Attach the missing keys to all inner nodes of the tree. (1 EC)

b) Insert the element 3 into this tree using the regular insert23 method. Update the keys. (2 EC)

c) In the tree resulting from b), delete 10. Update the keys. (2 EC)



Assignment 8: (5 EC)

a) Which is the exact task of a 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 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)? (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 9: (6 EC)

a) What is the run time of the algorithm of Floyd-Warshall for the all pairs shortest path problem? Denote the complexity class for n nodes. (1 EC)

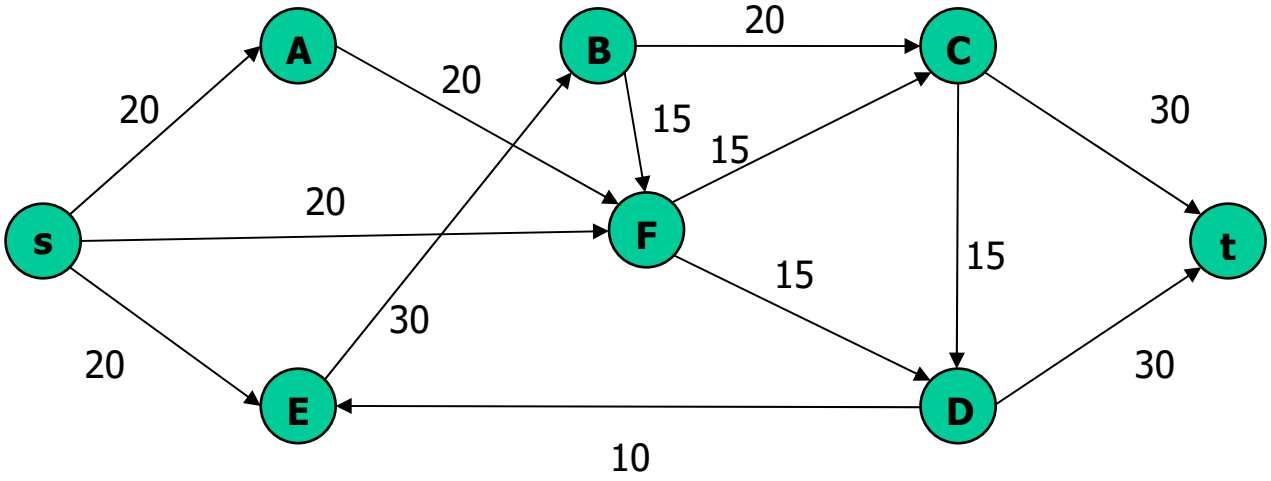
b) Specify the condition when Floyd-Warshall is advantageous compared to an iterative solution with Dijkstra's algorithm when each node is taken as a source subsequently. Compare the run time classes explicitely, and consider the same question for the case when iterative Dijkstra is better. (2 EC)

c) How can the all pairs shortest path problem be solved by matrix multiplication algorithms? Explain what is stored in the matrices A and A^k, and how the two matrix operations multiplication and addition are redefined. (2 EC)

d) Can Strassen's improvement to matrix multiplication be used for improving the all pairs shortest path problem? Justify your answer. (1 EC)

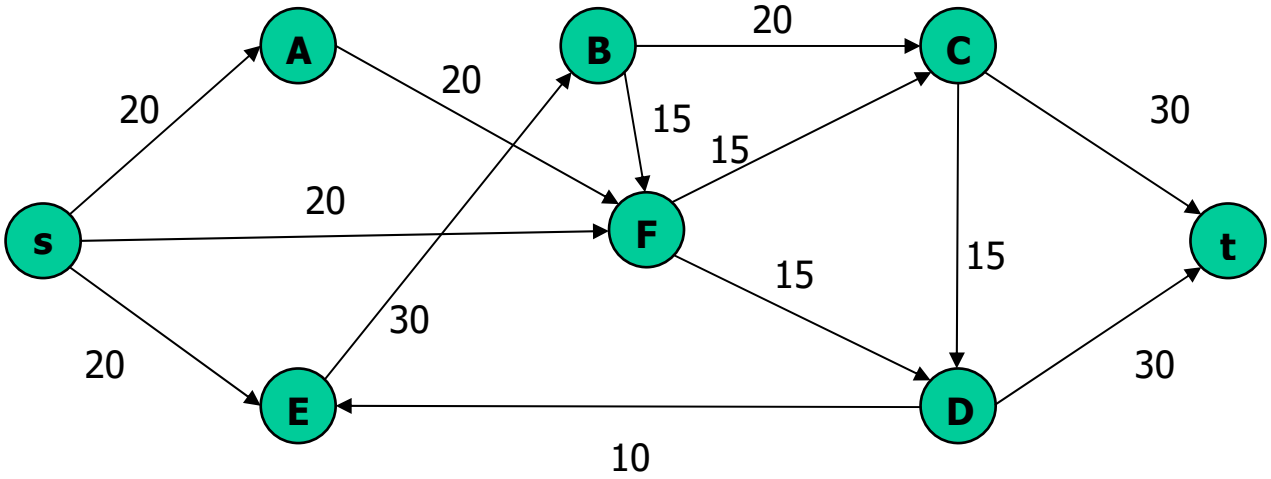
Assignment 10: (6 EC)

Consider the following graph with the given flow capacities:



a) In the above graph, show the result of the first iteration of the algorithm of Edmonds-Karp in order to compute the maximum flow in the above graph. What is the value of the resulting flow? (2 EC)

b) In the graph below, show the result of the first iteration of the algorithm of Dinic. What is the value of the resulting flow? (4 BE)



Assignment 11: (4 EC)

a) Define the prefix function of the algorithm of Knuth-Morris-Pratt by explaining in words how the output is related to the input. Of course, you must also mention the data types of input and output and how these values are related to the pattern which should be analyzed. (2 EC)

b) Illustrate the prefix funtion by giving the result for the following pattern: (2 EC)

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	b	b	a	b	b	b	a	b	b	a	b	b	a	b	b	b	a	b	b	a	b	b	b

.

Assignment 12: (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. (2 EC)