Final exam Algorithmics SS 2021

Prof. Dr. Sebastian Iwanowski 25.08.2021

Hints:

Time limit: 120 minutes

Admitted appliances: none (calculator allowed if needed)

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

Good luck!

Assignment 1: 3 EC

Sort the following complexity classes by inclusion:

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

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

c) $O(n^2)$

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

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

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

g) $O(\mathrm{sqrt}(n) (\log_2 n)^{100})$

h) $O(\mathrm{sqrt}((n^4 (\log_2 n)^3))$

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 h).

Assignment 2: 6 EC

Analyse Selectionsort:

a) Give the recursive formula for the **lower** run time bound if you want to sort n items.(1 EC)

b) Define the best complexity class for this lower run time bound and prove this formally by mathematical induction using a). (4 EC)

c) If you got a lower run time bound of $\Omega(f(n))$ for some $f(n)$, argue why Selectionsort satisfies a stronger criterion than needed for the Ω behaviour. You may answer this question by comparing the definition of Ω with what you can prove for Selectionsort.(1 EC)

Assignment 3: 4 EC

Consider Quadratic Binary Search (QBS):

a) Give a recursive formula for the upper bound of the run time of QBS at worst case. (1 EC)

b) Which is the best complexity class this run time of QBS belongs to ? (1 EC)

c) Argue why the complexity class of regular binary search is better and justify why QBS may still be preferred in practice. (2 EC)

Assignment 4: 5 EC

Consider the problem to apply the dictionary operations for names of at most 20 literals as keys. Consider the scenario where, first, 1000 names are inserted in a sequence, and afterwards certain names are searched for.

Take the name's length (i.e. the number of leading literals that are nonblanks) as a hash function and use closed hashing, i.e. hashing with linked lists at the index of the hash function.

a) How many comparisons do you approximately expect in average for the above scenario? (2 EC)

b) What is the worst case that may happen, and how many comparisons do you expect for this case? (1 EC)

c) What is the answer for a) and b) when you insert the 1000 names into a (2,3) tree with lexicographic ordering? (2 EC)

Assignment 5: 7 EC

a) Which is the exact task of a union-find algorithm defined for arbitrary sets? Specify the two functions provided by giving the input and the output in general. (2 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 in the most efficient way. Then show, how the first and the second set are united. Finally, show how the result is united with the third set. (3 EC)

c) 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. (2 EC)

Assignment 6: 4 EC

Consider the algorithm of Dijkstra:

- Initialize the node set **Done** by s;
  Initialize the node set **Undone** by all other nodes of graph G;
  For all nodes v of the graph G:

  Let label (v) := length of edge from s to v (∞ if no edge is existing, 0 if v = s);

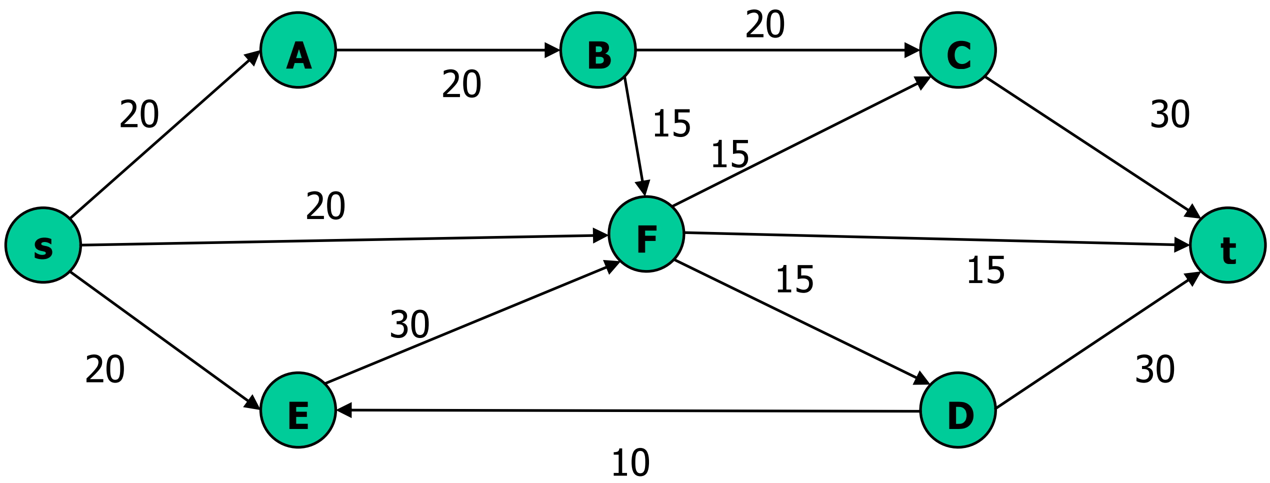
- While **Undone** is not empty:
  Search and delete the node v from **Undone** with minimal label;
  Insert v into **Done**;
  Update all neighbors n of v that are in **Undone**:
  If label (n) > label (v) + length of edge between v and n:
  Replace label (n) by that number;
  Let v be the predecessor of n.

Give the best complexity class for the worst case run time for an arbitrary graph with n nodes and nonnegative edge costs.

Justify this class by annotating the run time in the single steps of the above algorithm.

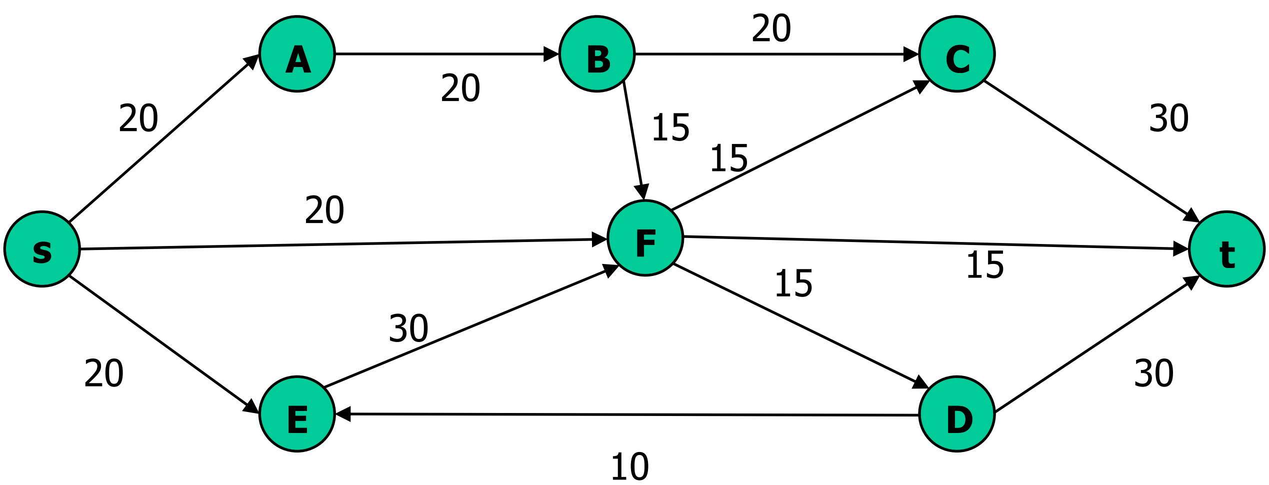
Assignment 7: 6 EC

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? (1.5 EC)

b) In the graph below, show the result of the first iteration of the algorithm of Dinic. You should also cross out edges that are not considered in the level graph. What is the value of the resulting flow? (2.5 EC)



c) Which edge of the graph in this assignment should be deleted such that the algorithm of Dinic finds a much better flow in the first iteration than the algorithm of Edmonds-Karp. Answer and justify here: (2 EC)

Assignment 8: 3 EC

Consider the string matching algorithm of Knuth-Morris-Pratt

a) Illustrate the prefix funtion by giving the result for the following pattern: (1 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	a	b	b	a	b	b	a	b	a	b	b	a	b	b	b	a	b	a	b	b	a	a

b) Consider the code of the main procedure:

```
i := 1; q := 0;
while i ≤ n do
{
    while (q>0)and (T[i] ≠ P[q+1])
        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;
}
```

Why is the run time of this procedure in O(n), although there is a nested loop in there? Justify this by giving the decisive arguments. (2 EC)

Assignment 9: 5 EC

Consider the problem **Largest Empty Circle**:

Given n reference points in the plane. Search for three points whose circumference does not contain any other reference point and has the largest radius with this property.

a) Solve this problem with a trivial algorithm: Describe it in words. What is the run time? (3 EC)

b) Assume that a Voronoi diagram is already computed and use this to solve this problem. What is the run time now? (2 EC)

Assignment 10: 7 EC

Consider the features of the plane sweep algorithms for Closest Pair and Voronoi (algorithm of Fortune):

a) What is the crucial difference between the two algorithms? (2 EC)

b) Both algorithms show an O(n log n) worst case run time. Answer the following questions:
i) What is the run time for the insertion of a new event?
ii) What is the run time for the update of the sweep status structure? (2 EC)

c) Describe a typical **dynamic** (i.e. after the initial events have been computed) event insertion for both algorithms respectively. (2 EC)

d) Describe a typical sweep status structure update for both algorithms respectively.(1 EC)