Final exam Algorithmics SS 2022

Prof. Dr. Sebastian Iwanowski 31.08.2022

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) plus 2 bonus credits.
For passing this exam you need at least 25 EC.

Good luck!

Assignment 1: 3 EC

Discuss the theoretical basics of Algorithmics by answering the following questions:

a) In general, why is it more important for practice to improve asymptotic runtimes for the underlying algorithms than to improve a software using a more efficient programming language in connection with an efficient compiler? (1 EC)

b) What is the difference between the O and Ω definitions which make the concept not totally symmetric? Explain why this may lead to different notions of the Θ definition. (2 EC)

Assignment 2: 4 + 2 EC

Analyse Quicksort:

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

b) Define the best complexity class for this upper run time bound and show the first line of the proof where the inductive assumption is applied to the recursive formula. (2 EC). If you complete this poof, you get up to 2 bonus EC (no withdrawal of credits of you fail in the bonus task). (2 + 2 EC)

c) Mergesort has got a better upper run time bound. Give two different arguments why Quicksort may still be preferred in practice. (1 EC)

Assignment 3: 5 EC

Apply the worst case optimal select algorithm for the following input:

Select (7, A) für A = 23 3 24 4 25 5 27 7 29 8 31 9 30 10 29

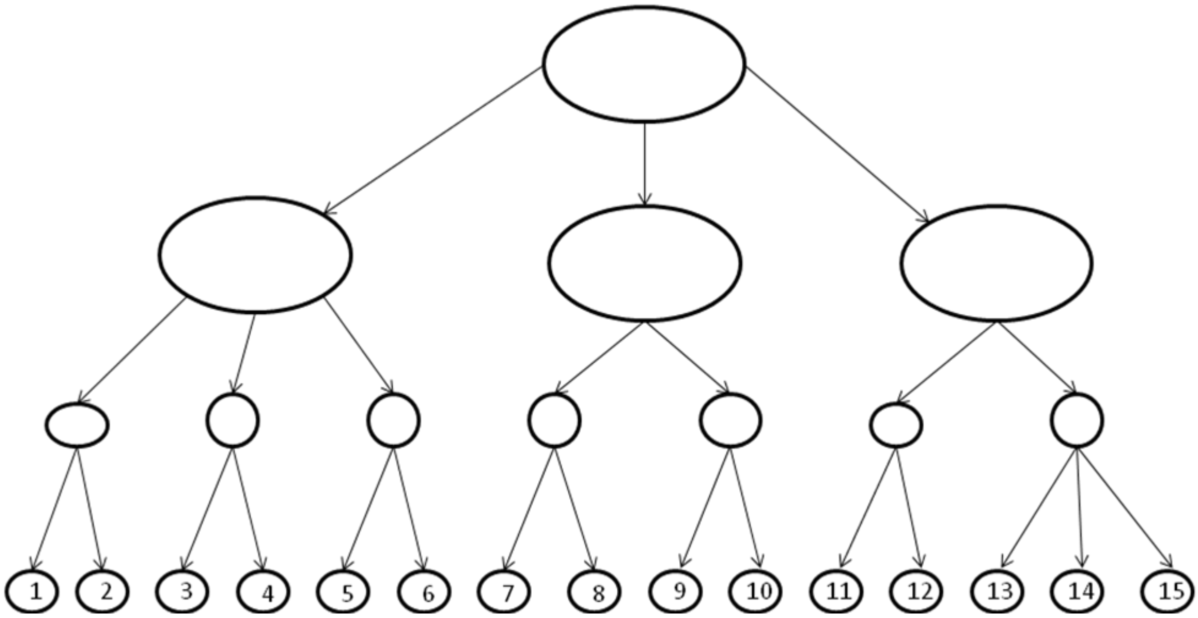
Choose the constant c=3 for the nonrecursive call.

a) Describe the first steps of the algorithm by giving the intermediate results. You may finish your description with the invocation of the first recursive call. (4 EC)

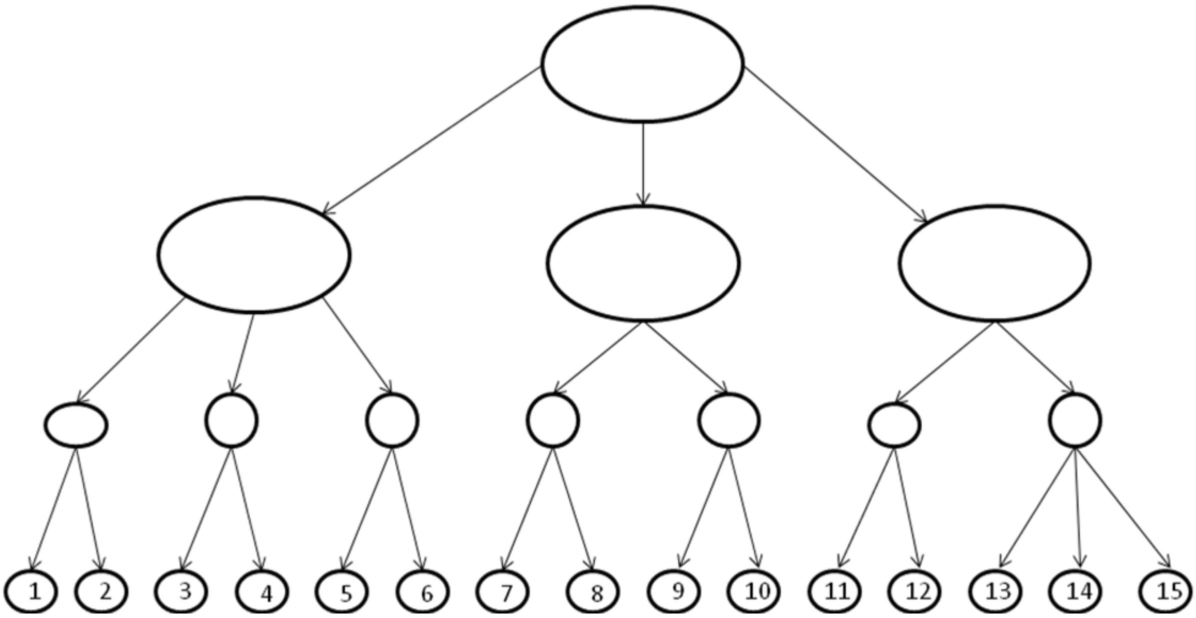
b) In your description, you have to split the input array in some way. Comment on the size of the split: May it be arbitrary? Which other sizes are you allowed to take without deteriorating the asymptotic worst case run time behavior? (1 EC)

Assignment 4: 4 EC

a) In the 2-3 tree below, simulate a delete procedure that distributes the grandchildren to new children in a recursive step by deleting element 4. Delete deleted nodes and insert newly created nodes into the picture together with the new connections. Furthermore, give the values of the keys in the first two levels. (2 EC)



b) Simulate the insertion of element 16 in the 2-3 tree below in the same way as in a). Also enter the keys of the changed tree again. (2 EC)



Assignment 5: 3 EC

Consider the algorithm of Bellman:

```
for i = 0,...,n do
    w_{i+1,i} = q_i
    c_{i+1,i} = 0
end for
for k = 0,...,n-1 do
    for i = 1,...,n-k do
        j = i + k
        Determine m where i ≤ m ≤ j, s. that c_{i,m-1} + c_{m+1,j} Is minimal
        r_{i,j} = m
        w_{i,j} = w_{i,m-1} + w_{m+1,j} + p_m
        c_{i,j} = c_{i,m-1} + c_{m+1,j} + w_{i,j}
    end for
end for
```

a) What problem does this solve? Describe the input and the output for this algorithm. (2 EC)

b) Consider the following table produced by the algorithm of Bellman:

i	0	1	2	3	4
init	$w_{1,0} = 0{,}1$ $c_{1,0} = 0$	$w_{2,1} = 0{,}1$ $c_{2,1} = 0$	$w_{3,2} = 0$ $c_{3,2} = 0$	$w_{4,3} = 0{,}1$ $c_{4,3} = 0$	$w_{5,4} = 0$ $c_{5,4} = 0$
k=0		$r_{1,1} = 1$ $w_{1,1} = 0{,}4$ $c_{1,1} = 0{,}4$	$r_{2,2} = 2$ $w_{2,2} = 0{,}2$ $c_{2,2} = 0{,}2$	$r_{3,3} = 3$ $w_{3,3} = 0{,}2$ $c_{3,3} = 0{,}2$	$r_{4,4} = 4$ $w_{4,4} = 0{,}4$ $c_{4,4} = 0{,}4$
k=1		$r_{1,2} = 1$ $w_{1,2} = 0{,}5$ $c_{1,2} = 0{,}7$	$r_{2,3} = 2$ $w_{2,3} = 0{,}4$ $c_{2,3} = 0{,}6$	$r_{3,4} = 4$ $w_{3,4} = 0{,}5$ $c_{3,4} = 0{,}7$	
k=2		$r_{1,3} = 1$ $w_{1,3} = 0{,}7$ $c_{1,3} = 1{,}3$	$r_{2,4} = 3$ $w_{2,4} = 0{,}7$ $c_{2,4} = 1{,}3$		
k=3		$r_{1,4} = 2$ $w_{1,4} = 1$ $c_{1,4} = 2{,}1$			

Give the final output of the algorithm of Belman. (1 EC)

Assignment 6: 5 EC

Consider the efficient Union-Find algorithm:

a) Describe the interface of the two functions Union and Find: What is the input for the function and what is the output. (2 EC)

b) Denote the asymptotic run time for both functions. (1 EC)

c) In efficient implementations, the data of the Union-Find algorithm is stored in an array (not tree). What is the content at a certain position of this array? Mention all data necessary. (1 EC)

d) Give a graph application where the Union-Find algorithm is crucial in order to maintain the optimum run time. Denote the problem to be solved and the name of this algorithm.(1 EC)

Assignment 7: 6 EC

Consider the algorithm of Floyd-Warshall for the All Pairs Shortest Path problem:

for $i = 1, \ldots, n$ **do**
  for $j = 1, \ldots, n$ **do**
    $d_{ij}^{(0)} = \begin{cases} c(i,j): & \text{falls } (i,j) \in E \\ \infty: & \text{sonst} \end{cases}$
  end for
end for
for $k = 1, \ldots, n$ **do**
  for $i = 1, \ldots, n$ **do**
    for $j = 1, \ldots, n$ **do**
      $d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$
    end for
  end for
end for

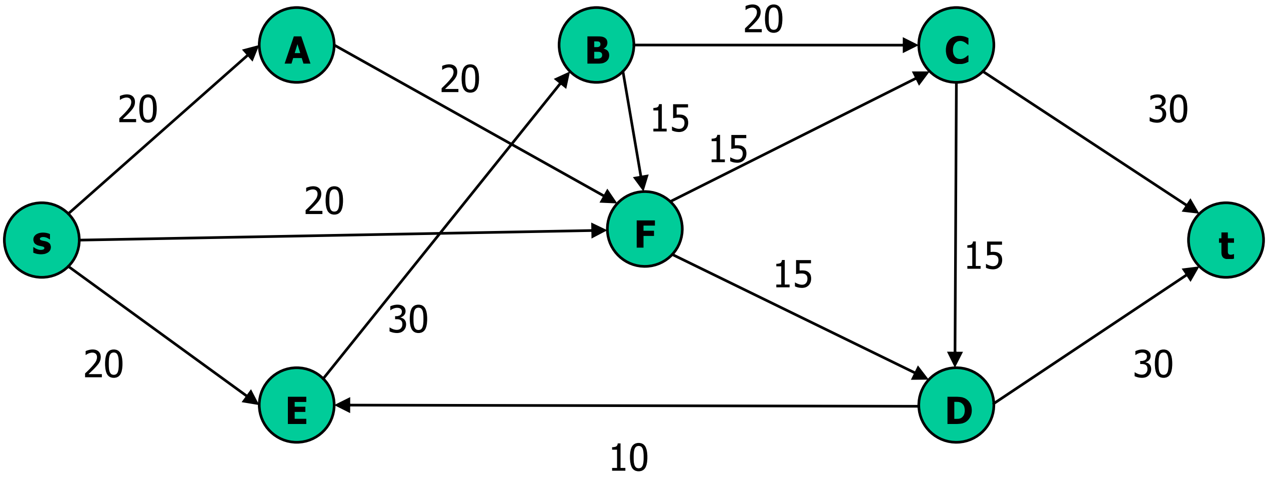
a) Prove the correctness of this algorithm by mathematical induction using one of the above parameters. (3 EC)

b) Denote the asymptotic run time of this algorithm and justify this in the above code. (2 EC)

c) Denote a scenario where this algorithm has an advantage to the repeated application of Dijkstra. Justify your answer. (1 EC)
<u>Hint:</u> Besides dedicated run-time comparisons you may answer this question also by denoting a scenario where Dijkstra's algorithm cannot be used at all.

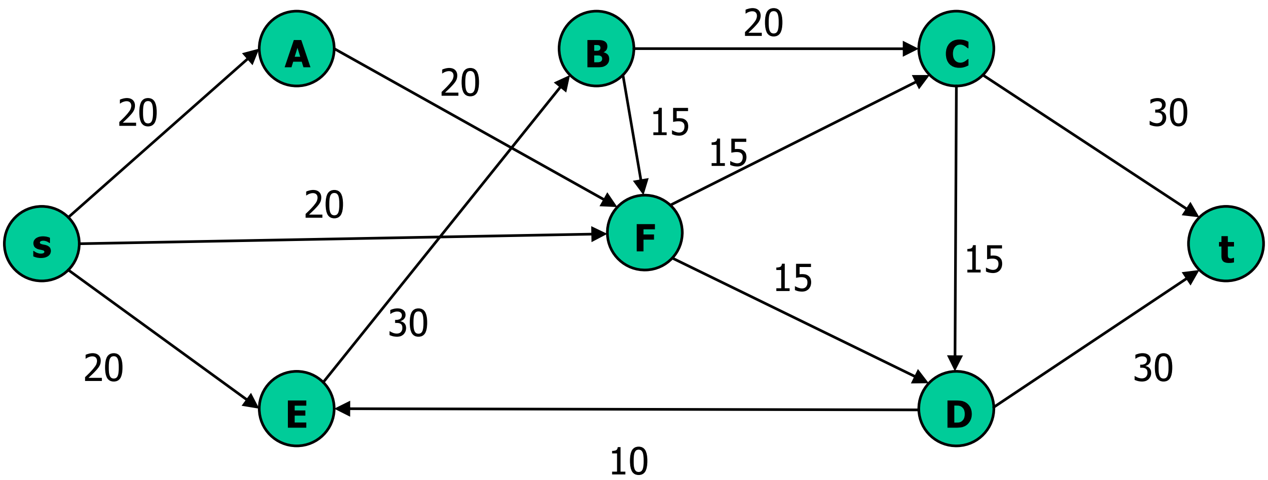
Assignment 8: 4 EC

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)



Assignment 9: 5 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	b	a	b	b	b	a	b	b	a	b	b	a	b	b	b	a	b	b	a	b	b	b

b) Explain by words how a proof works with an amortisation argument. You need not illustrate this by an actual example. (2 EC)

c) Using the amortisation argument what is the run time of Knuth-Morris-Pratt? What would be the run time of the trivial algorithm. In your answer, explain the meaning of all parameters. (2 EC)

Assignment 10: 6 EC

Consider the plane sweep algorithm for Closest Pair:

a) Describe the two different insert types into the Event Point schedule and distinguish which Insert step is performed at the beginning of the algorithm and which one dynamically during the course of the algorithm. (2 EC)

b) Why does the insertion of an event cost O(log n) time? Justify why this cannot be performed faster such as in constant time and mention which decisive geometric property guarantees that this does not cost considerably slower time. (2 EC)

c) Using a) and b), denote and justify what is the total time of the algorithm. (2 EC)

Assignment 11: 5 EC

Consider the problem V to compute a Voronoi diagram:

a) Denote a problem A which justifies why V cannot be solved faster than in O(n log n) time (just call the name of A). (1 EC)

b) What is the reason for the lower bound of A? As answer, denote a third problem B which may be solved in a time faster than O(n log n) using A. The way how this is done need not be mentioned here. (1 EC)

c) Sketch how A can be solved using V in a time faster than O(n log n). (3 EC)