

Resit Exam ADS, I, Module 7, Codes 202001360 & 202500367

Algorithms and Data Structures

Monday June 15, 13:45-15:45, 2026

All answers need to be motivated! You are allowed to use a handwritten cheat sheet (A4, both sides) during the exam.

Total amount of points 90, grade: $0.1x + 1$ where x is the number of points obtained.

Repeat students: you do not need to do the starred exercise 3. Number of grade points for this exam: $0.05x$ where x is the number of points obtained (so a maximum of 4 grade points; you get a grade for the "old" ADM course component by also doing the ADM exam for which you can get an additional 5 grade points plus 1 extra point).

1. (20 points)

1(a). Given following algorithm where $//$ is integer division (so $5//2=2$):

```
def fun(n):  
    if n <= 1:  
        return 1  
    a = fun(n // 2) + fun(n // 2)  
    b = 1  
    for i in range(n):  
        b = b*i  
    return a + b
```

Let $T(n)$ be the number of multiplications for $\text{fun}(n)$. Give a recursive equation for $T(n)$.

1(b). Suppose that the number of steps of an algorithm $T(n)$ with an input of n , has the recurrence relation

$$T(n) = 3T(n/3) + \sqrt{n}$$

What is the asymptotic complexity class of this algorithm?

2. (30 points) Do you agree with the following statements? If "yes", explain why, if "no", give a counterexample.

2(a). If you reverse the order of the elements in a max heap array you get a min heap.

2(b). If a binary tree (with possibly duplicate elements) is ordered in an in-order way, it is a binary search tree.

2(c). Searching for an element in a binary search tree with n elements has asymptotic time complexity $\Theta(\log(n))$.

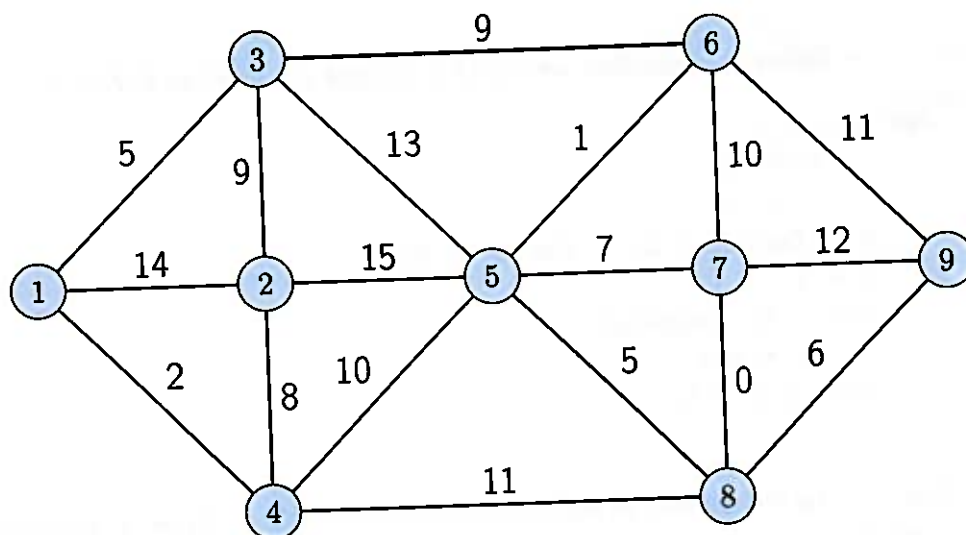
*3. Not for repeat students! (10 points)

Prove the following proposition using induction:

For all $k \geq 0$: a tree of height k has at most $2^k - 1$ nodes that are left children.

4. (10 points)

Given the following weighted graph:



Determine the minimum spanning tree using Prim's algorithm; indicate the order in which you find the edges (start at vertex 1).

5. (20 points)

Given a $M \times N$ matrix where each cell has a cost associated to it. We are interested in the minimum cost to reach cell $(M-1, N-1)$ starting from cell $(0,0)$ where you can only move one unit right or one down from any cell, i.e. from cell (i,j) you can move to $(i,j+1)$ or $(i+1,j)$.

5(a). We want to fill a matrix T such that $T(i,j)$ contains the minimum cost for going from $(0,0)$ to (i,j) . Explain that for $j > 0$, $T(0,j) = \text{cost}(0,j) + T(0,j-1)$, and for $i > 0$, $T(i,0) = \text{cost}(i,0) + T(i-1,0)$. Give an expression for $T(i,j)$ for $i, j > 0$.

- 5(b). Give an algorithm (using dynamic programming) that gives the minimum cost for going from $(0, 0)$ to $(M - 1, N - 1)$. The algorithm should have complexity not bigger than $\Theta(MN)$.

