material The material for the exam is everything which has been discussed in class (both lecture and exercise class) in 2016-2017. See for example the webpage. The list below may be helpful to organize the material.

data structures
D1 linear data structures:
    stack, queue, priority queue, singly linked list, doubly linked list
    hash table
D2 trees:
    binary tree, (min- and max-)heap, binary search tree, AVL-tree
D3 graphs:
    simple, connected, directed or undirected, weighted or non-weighted

classifiers
A1 sorting:
    insertion sort, selection sort, bubble sort, merge sort, heapsort, quicksort
    counting sort, radix sort, bucket sort
A2 traversing:
    tree traversals
    (Graph traversals were not discussed in 2016-2017 and hence are not part of the material for the exam.)
A3 varia:
    greedy algorithms:
    for example activity selection, fractional kanpsack, Huffman, Dijkstra’s shortest path
    dynamic programming algorithms:
    for example Fibonacci numbers, max-sub-array, rod cutting, knapsack01, LCS, all pair shortest paths
    (The dynamic programming algorithm for matrix multiplication was not discussed in 2016-2017 and hence is not part of the material for the exam.)
    other algorithms mentioned in class
A4 string matching algorithms:
    brute force string matching algorithm, Knuth-Morris-Pratt string matching algorithm
    (The Boyer-Moore string matching algorithm was not discussed in 2016-2017 and hence is not part of the material for the exam.)

complexity analysis

C1 worst-case time complexity
C2 $O$
C3 $\Theta$
C4 amortized complexity
C5 lower bound comparison-based sorting using $\Omega$
C6 paradigms: divide and conquer, dynamic programming, greedy

pseudo-code The exam will contain at least one question where pseudo-code is asked. Pseudo-code might be asked of a ‘new’ (in the sense of not discussed in class) algorithm. It is not the intention, and will hardly be useful, to memorize the pseudo-code of all algorithms that we discussed. In particular, pseudo-code of algorithms for linear sorting, AVL-tree rotations, LCS, KMP-string-matching will not be asked. The two old exams and two old midterms give an idea of the amount of pseudo-code that is asked usually.