EXAM RULES:



* This is a take-home exam, which means that you can use all and any study materials you need.
* Please complete this exam by yourself. It is expected that you will work by yourself and will not ask neither receive help for the exam from any other students or individuals.
* Prepare your answers on separate sheet(s) of paper. You can either write your answers down by hand or prepare them on a computer and then print and submit that. Write your *full name* and *student id* on your answers!

Good luck!

**Q1:** Solve the recurrence using the Master Theorem.

**Q2:** Show that the solution of the recurrence is   
 using the recursion-tree.

**Q3:** Solve the recurrence using any method.

**Q4:** Airline operators routinely oversell flights because they observe that always there are passengers who do not appear for the flight. Consider a flight with 300 passengers and assume that the probability that a passenger does not appear for the flight is 5%, and the passengers do so independently. Use the statement from probability known as the Central Limit Theorem to show that the probability of at least 5 people not appearing for the flight is at least 99.9%.

**Q4:** Consider an array of numbers . Write an algorithm that finds the kth smallest number in (the kth order statistic) in at most time.

**Q5:** Consider two sorted arrays of numbers and . Write an algorithm that finds the set-difference of the two arrays in at most time. (The set difference of *A* and *B* is an array consisting of all elements that are not in *B*.)

**Q6:** In our discussion of the search algorithm for the Interval Binary Search Trees in the class,

Algorithm Seartch-IT(*T*,*t*)

while and not ()

if

else

return *x*

the Loop Invariant “If there are any intervals in *T* thatcontain *t*, then at least one such interval is in the subtree of *T* with the root at node *x-*  *.*” was used. As the means of proving that this Loop Invariant is true for every iteration of the while-loop, show that: If there is any interval *int* containing *t* in the subtree then, (i) if , where the maximum is over all intervals in the left subtree , then there must be at least one such interval in the left subtree ; (ii) if , then any such intervals must be in the right subtree . (Hint: Recall that for all intervals in , , because *T* is a binary tree ordered on .)

**Q7:** Write the algorithm for a dynamic programming solution of the integer knapsack problem discussed in the class. The integer knapsack problem is: for a set of objects of quantity , cost , and size each and a constant *S* – the knapsack size – choose a set of integer allocations such that is maximized while also . (Hint: First replace the quantity of each object *i* with objects of quantity 1, so that at each step you only need to decide whether to take an object or not – or , to simplify you problem. Then, imagine that you know that one object is in the optimal knapsack; consider how the remaining objects need to be chosen then, in order to write down a reduction relationship. Use the reduction relationship to write a top-down, recursive, memorization dynamic programming solution algorithm.)