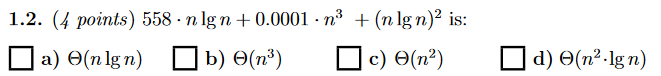
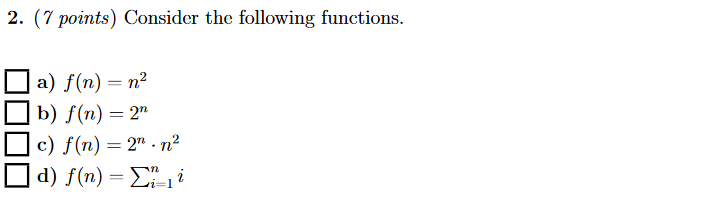


b) because

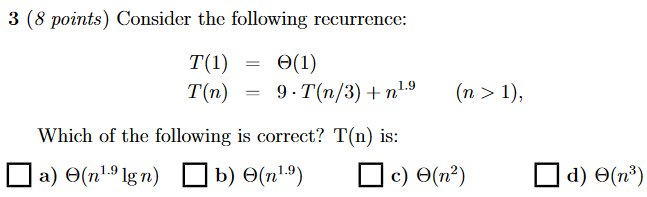


b)

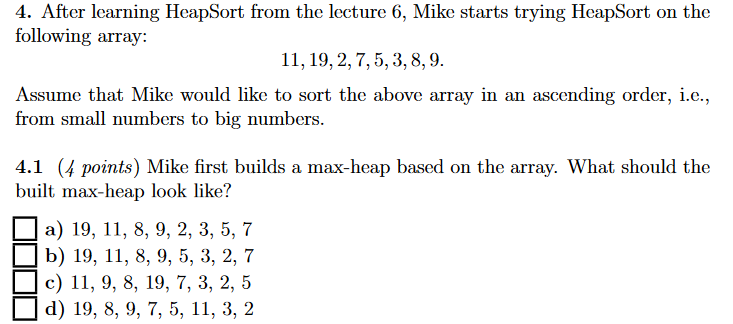




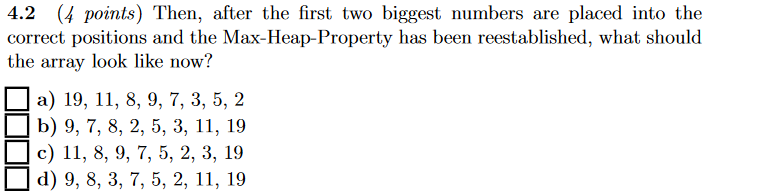
1. d) because d is the same as which is the same as , which is = to which is the same as because we do not care about constants when it comes to growth.



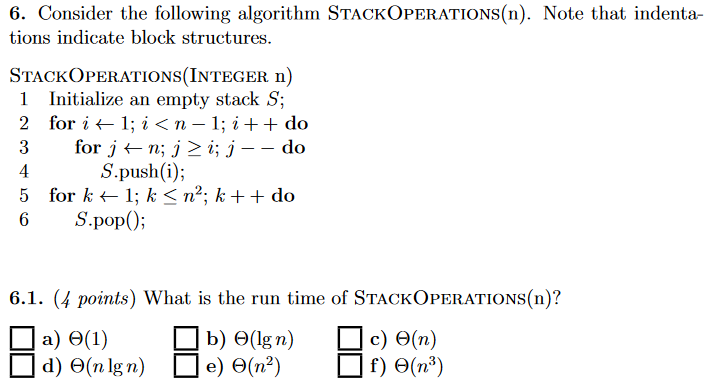
1. see lesson 4 notes



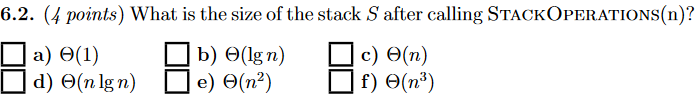
b)



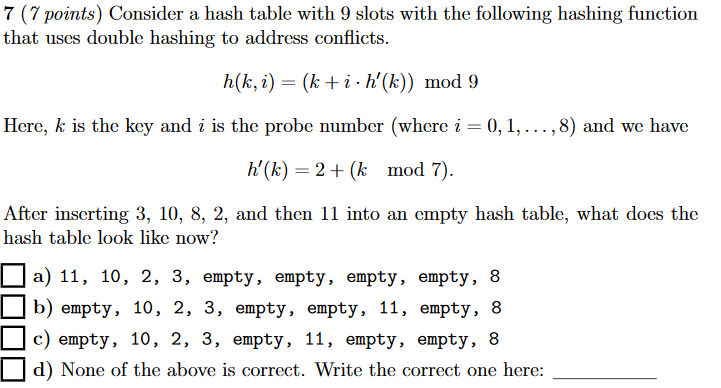
b)



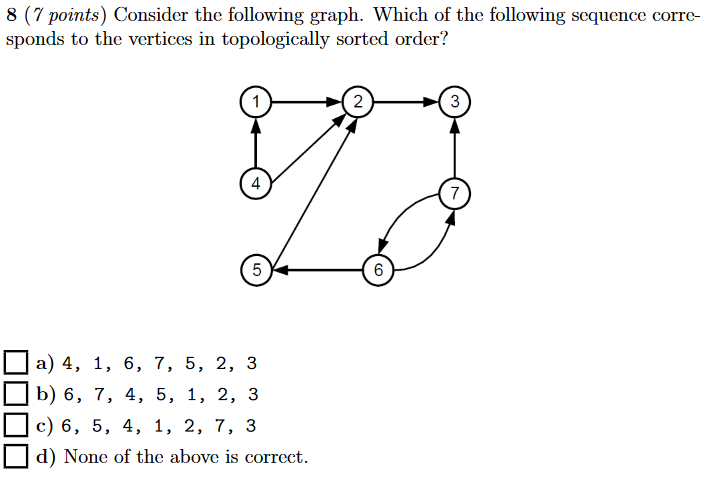
e)



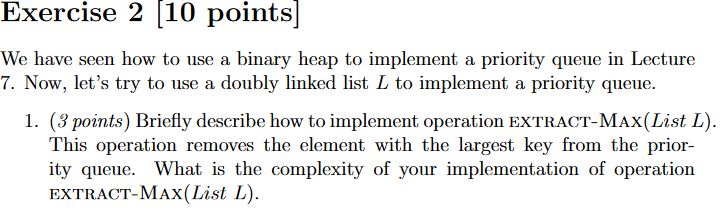
a)



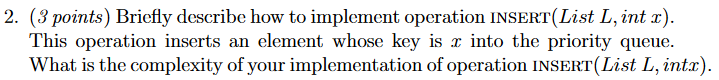
c)



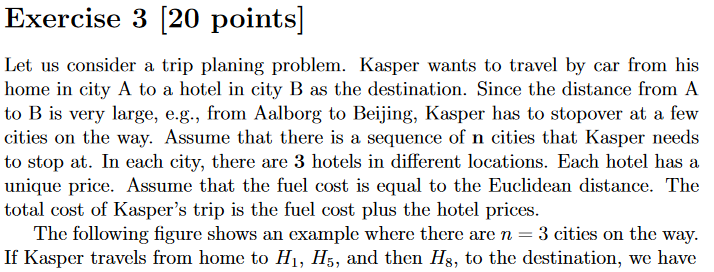
d – because cycle between 6 and 7

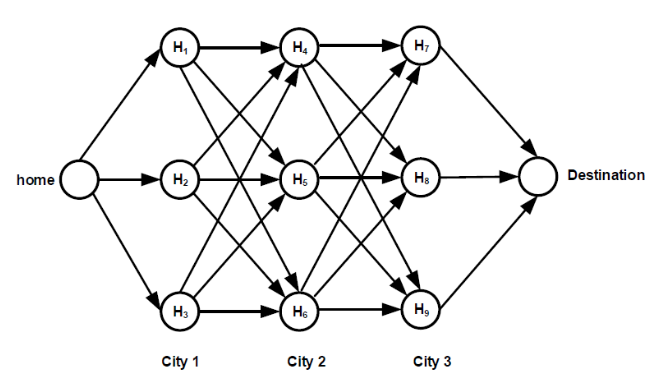


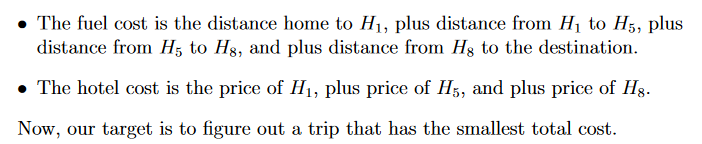
Since list L is a priority queue, the greatest value is always at the top. This means that with extract-max we just need to get the first element in L.

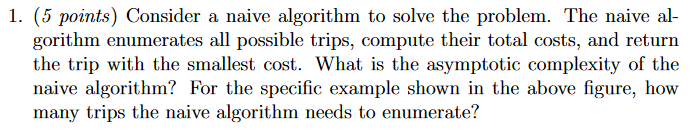


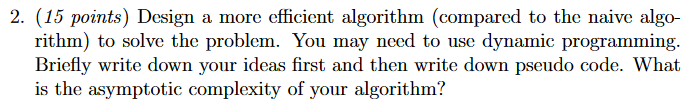
Go through the entire list to find a set of 2 numbers where one is great and another is lesser than x. Then place x between those two numbers. Worst-case runtime is n.



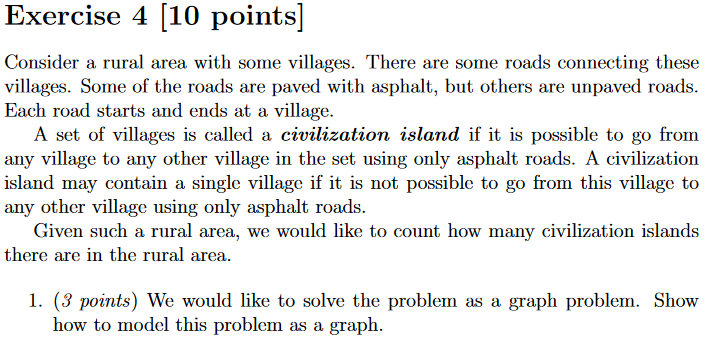








Use Dijkstra’s algorithm, take into account the travel cost to a hotel plus the hotel cost.



Mark villages as vertices, use asphalt roads as edges.



Depth-first search boi.