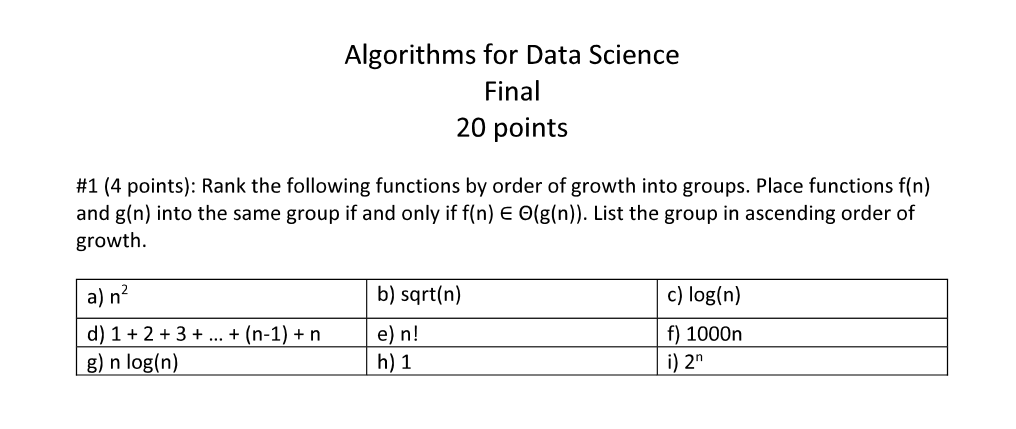
Ethan Engel

Summer 2021



1. *A) 1(no growth) B) log(n) C) sqrt(n) D) 1000n*

*E) nlog(n) F) & (1 + 2 + 3 + 4 + . . . .n) = G) H) n!*

Graphical user interface, text

Description automatically generated

1. *Keys for hash tables come from the immutable domain, of which lists are not a member. Even if it were possible, it would be impractical, because lists cannot easily be used in hash functions the way that single integers can.*

#3 (4 points) We covered several data structures: list, stacks, queues, heaps, and hash tables. For each problem below choose the best data structure that can be used to solve the problem efficiently. (a) Given a list A of n integers, reverse the order of the elements in A. (b) Given a list A of n integers, report the ten largest integers in A. (c) Given an undirected graph G determine if G is connected.

1. *A) Only one iteration is necessary, without checking for any particular traits of the data, so a normal list would be efficient.*

*B) A (maximum) heap would keep the largest of a group of integers accessible for removal, thus would be the most efficient option.*

*C) A depth first search would be employed, to see if all the nodes could be visited from an origin node. Stacks are used for DFS.*

#4 (5 points): We covered several algorithm design techniques: incremental, divide-and-conquer, dynamic programming, greedy, and backtracking. For each problem below choose the most efficient technique that can be used to solve the problem efficiently. If more than one choice is available select the one that is simplest to implement. Justify your answer.

1. Given a list of n triangles, find the range (i.e., min and max) of the perimeters of the input triangles.

(b) Sort a list of n triangles in decreasing order of perimeter.

1. Given a list of n items and their corresponding integer weights, maximize the total weight of items that fit in a backpack of capacity W by selecting items from the given list.

(d) Given a list of n items and their corresponding integer weights, maximize the total number of items, selected from the given list, that fit in a backpack of maximum capacity W.

1. *a) We should divide and conquer the array into 2 arrays, finding the maxima & minima of each half, which would then be combined into one overall range.*

*b) For sorting, a DAC algorithm, like mergesort would be most efficient.*

*c) This is essentially a bin-packing problem with one bin, which uses incremental techniques to maximize weight per bin.*

1. *If what we want to do is maximize the number of items, a greedy algorithm, which simply chooses the lightest item each time, would be employed.*

#5 (4 points): Explain the similarities and differences between Divide-and-Conquer and Dynamic Programming. When would you use Dynamic Programming instead of Divide-and-Conquer?

1. *Divide & Conquer algorithms break down a problem into fractions of the original size (n). These fractional problems are then solved. These solutions are examined and combined to determine the final solution. Dynamic Programming breaks down a problem into smaller problems as well, but the earlier problems results are memorized. In this manner, if a route of a subproblem is not optimal, then there is no need to progress further down its path, as it cannot lead to the overall optimum. The primary similarity is the partitioning of the original problem. The primary difference is the memorization of partial solutions in dynamic programming, or conversely, the independence of the fractional problems in DAC algorithms. If the subproblems overlap (are not independent), then dynamic programming should be used as it can reduce runtime from exponential to polynomial.*