# Sample exam algorithms and data structures 1

There are 100 points in total. Grade = points/100.

|  |
| --- |
| 1. Merge sort 2. (10 point) Apply merge sort to the letters below. Show all the steps of the sorting process.   A D X C F M P L |
| |  |  | | --- | --- | | Step | A D X C F M P L | | 1 | A D X C ||| F M P L | | 2 | A D | X C ||| F M | P L | | 3 | A | D || X | C ||| F | M || P | L | | 4 | A D|| C X ||| FM || L P | | 5 | A C DX ||| F L MP | | 6 | A C DFL M PX | |
| 1. (5 point) What is the worst-case time complexity of merge sort? |
| O(Log(n)) |
| 1. (5 point) Merge sort is a stable sorting algorithm. What does “stable” mean in this context? |
| Some of the items are staying in the same space, thus it’s easier to go back to the initial positions. |
| 1. Dating   A dating agency aims to help people to find a partner. They have a database with a large number of registered people looking for a partner. When registering with the agency a person needs to choose a number of interests from a list of about 1000 interests. The agency matches the interests of the new member against the interests of all other registered people. They need an algorithm to find the 5 people with highest number of overlapping interests as potential partners for the new member.   1. (13 points) Create a data structure for storing people’s interests and an efficient algorithm for finding potential partners. Describe the data structure(s) used and give the algorithm in pseudocode or Java code. |
| ~~Can use an array of adjacency list because it is best used for a sparse graph as a data structure. As for an algorithm can be used Breadth First Search, which .~~ |
| 1. (7 points) What is the worst-case time complexity of your algorithm? Explain your answer. |
| ~~O(N~~~~2~~~~) – have to check whole list~~ |
| 1. Bags and sets 2. (3 points) What is the difference between a Bag and a Set? |
| |  |  |  |  | | --- | --- | --- | --- | | Bag | Set | Que | Stack | | * No order * Can add in anything (ie: int, Obj, String) * Can add duplicates | * No duplicates allowed * Only one type * Can be ordered |  |  | |
| 1. (4 points) Which operations are by default available for a Bag? |
| add – adding  no – removing  IsEmpty - Boolean  Size() - int |
| 1. (4 points) The Java Collection Framework provides two types of sets: TreeSet and HashSet. Under which circumstances is it better to use a TreeSet than a HashSet? |
| TreeSet – elements sorted in ascendant order by default; can access the 1st or last element  HashSet – does not sort elements, it’s much faster; cannot access the 1st or last element |
| 1. (4 points) Give two alternative ways to implement a Bag. |
| Array  LinkedList - |
| 1. Expressions often contain multiple sets of parenthesis of various types (round, square, and curly brackets). If the parentheses do not match the expression is invalid. For example, the following expression is valid:   {(4 + a) \* ([b + 3] \* 7)}  The following expressions are invalid:  ((a + b) \* 4  ((a + b) \* 4]  {6 + a}) \* 7 (    The **incorrect** algorithm on the next page attempts to check whether an input string has matching parenthesis. It represents the input expression A as an array of symbols. The algorithm is shown in pseudocode.  ---------------------------------------------------------------------------------------------------  Input array A  int open\_square\_brackets = 0  int open\_curly\_brackets = 0  int open\_round\_brackets = 0  for (int i=0; i<A.length; i++)) {  if (A[i] == “[“ ) {  open\_square\_brackets++  } else if (A[i] == “{“ ) {  open\_curly\_brackets++  } else if (A[i] == “(“ ) {  open\_round\_brackets++  } else if (A[i] == “]” ) {  open\_square\_brackets--  } else if (A[i] == “}” ) {  open\_curly\_brackets--  } else if (A[i] == “)” ) {  open\_round\_brackets--  }  }  if (open\_square\_brackets == 0 and open\_curly\_brackets == 0 and open\_round\_brackets == 0) {  return true  }  return false   1. (10 points) What is the worst-case time complexity of this algorithm? Explain your answer. |
| O(N) – iterates over all the elements in the array + basic operations which have a complexity O(1) |
| 1. (10 points) The given algorithm is incorrect. For example, it will return true for the following invalid expressions:   (a + 2)) + b (  (a + 2] \* [3 + b)  Create an efficient correct algorithm for this problem: checking whether the parenthesis of a given expression match. Describe the data structure(s) used and give the algorithm in pseudocode or Java code. |
| ~~while( any open bracket exist ){~~  ~~Open bracket++~~  ~~Search for same type closing bracket~~  ~~If(closing bracket = exist){~~  ~~Open bracket--~~  ~~} if (closing brackets before open brackets){~~  ~~Throw exception;~~  ~~}~~  ~~else(closing bracket !=exist){~~  ~~Sys.out.print(“Error”)~~  ~~}~~  ~~if (open\_square\_brackets == 0 and open\_curly\_brackets == 0 and open\_round\_brackets == 0) {~~  ~~return true~~  ~~}~~  ~~return false~~    ~~}~~ |
| 1. (5 points) What is the worst-case time complexity of your algorithm? Explain your answer. |
| O(N2) |
| 1. Graphs 2. Directed Graphs can be used, for example, to represent dependencies between tasks (e.g. task A much be finished before task B can start). Give two examples of applications of directed graphs in different domains than task scheduling. |
| (2 points) Example 1  Links on a webpage |
| (2 points) Example 2  Flights (& destination airports) |
| 1. (5 points) What is a strongly connected component in a directed graph? |
| From any 1 node of a graph you can get to any other node in the graph |
| 1. (6 points) Apply Prim’s algorithm for finding the Minimal Spanning Tree of the following weighted graph. Show all steps of the algorithm and the final Spanning Tree.   2  20  4  8  1  10  12  5  9  6 |
|  |
| 1. (5 points) What is the worst-case time complexity of the lazy version of Prim’s algorithm for finding the minimal spanning tree? |
| O(ElogE) |