Exercise Session n. 9

Algorithms and Data Structures

Exercise 265 (m22)

Write an algorithm `Max-Heap-Insert(H, x)` that inserts a value x in a max-heap H. Also, write the content of H (as an array) after the insertion of each of the following values, in the given order, starting from an empty max-heap: 3, 7, 3, 2, 9, 5, 9, 8, 5, 2, 9, 4, 7, 3, 9

Exercise 267 (m22)

The following algorithm `Algo-Y(A, r, c)` operates on an r × c matrix of `n = r c elements`, where r and c are the numbers of rows and columns of the matrix, and the matrix is stored row-wise in the given array A. This means that the first c elements of A are the c elements of the first row of the matrix, the following c elements of A are the c elements of the second row of the matrix, and so on.

```
Algo-Y(A, r, c)
  for i = 1 to rc
    for j = i + 1 to rc
      if A[i] == A[j]
        a = ⌊(i - 1)/c⌋ // integer division
        b = ⌊(j - 1)/c⌋ // integer division
        if a == b or a == b-1
          if i - ac == j - bc or i - ac == j - bc + 1 or i - ac =
            return true
  return false
```

Question 1

Explain what `Algo-Y` does. Do not simply paraphrase the code. Instead, explain the high-level semantics of the algorithm independent of the code.

Question 2

Analyze the complexity of `Algo-Y`. Is there a difference between the best and worst-

case complexity? If so, describe a best and a worst-case input of size n, as well as the behavior of the algorithm in each case.

Question 3

Write an algorithm called Better-Algo-Y that does exactly the same thing as Algo-Y, but with a strictly better complexity in the worst case. Analyze the complexity of Better-Algo-Y.

Exercise 269 (m22)

We say that an array A is in "e-top" order when `A[i] ≤ A[j] for all i, j` such that i is odd and j is even. Write an algorithm `Sort-E-Top(A)` that sorts an array A in e-top order with an average-case time complexity of `O(n)`. You may want to use standard, well-known algorithms. However, you must explicitly write their pseudo-code.