Algorithms and Data Structures, Academic Year 2013/2014

International Bologna Master in Bioinformatics

December 16, 2014

Please complete the following exercises by applying the concepts that have been illustrated to you during the classes. The score associated with each exercise and the expected time for completion is reported in the first line. Do NOT copy/exchange results (the parameters of each exercise are different).

Exercise 0 (2 points): write your name and surname in the first row of all the sheets you use.

Name:_____ Surname:_____

Exercise 1 (35 points, 60 minutes): please design the data structures and provide a high level description (e.g. pseudo-code) of the main components of the algorithm that you would implement to **efficiently** realize a function that takes in input two strings, meant to be very large, and outputs the longest suffix of the first string that is a prefix of the second string. The implementation will be considered efficient if its complexity is **strictly better** than O(n*m) (where n and m are the lengths of the two strings).

For the implementation of the solution, please provide a motivation for your design, and a sketchy discussion of average/worst-case complexity in space and computation.

If you reuse standard data-structures/algorithms, you do not need to re-implement them. However, remember to take in account their complexity when computing the one of your solution.

(use additional sheets for this exercise, including the back of this sheet)

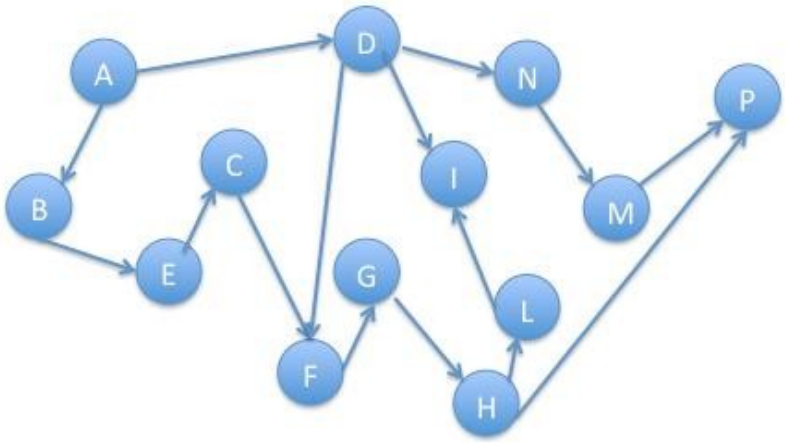
Name:_____ Surname:_____

Exercise 2 (18 points, 20 minutes): given the following sequences of visited nodes of a generic **binary** tree whose nodes are all **distinct**, write the tree itself in the space below. An “x” is a placeholder for an unknown number. Note: the solution is not unique.

pre-order visit: x 20 6 7 21 x x 11 5
in-order visit: x x x 12 x 16 21 5 11
post-order visit: 7 6 20 x 15 x x x 12

Name:_____ Surname:_____

Exercise 3 (15 points, 20 minutes): apply an algorithm to compute a topological sort of the graph below. Show the data structures involved at every major iteration of the algorithm. Assume that the nodes are stored in the adjacency set in alphabetical order.



Name:_____ Surname:_____

Exercise 4 (15 points, 10 minutes): Please compute the domain of this recursive function and its time and space complexity. Then determine the mathematical expression implemented..

```
function f(int x, int y)
begin
        if (x == y + 1) then return 1
        else return x + y + f(x + 1, y - 1)
end
```

Name:_____ Surname:_____

Exercise 5 (15 points, 10 minutes): compute the BWT and the LF map for the following string.

LA_BANANA_BALLA