

Algorithms and Data Structures, Academic Year 2013/2014

International Bologna Master in Bioinformatics

July 18, 2016

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 a large string **S** and a non negative integer **K** and outputs the length of the **longest suffix** of **S** that occurs **at least** **K** times. The string **S** can be **pre-processed** once and for all. Full score will be achieved only with an algorithm **linear** in the length **n** of the string **S** (i.e. in $O(n)$ time).

For the implementation of the solution, please provide a motivation for your design, and a sketchy discussion of 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 with **repetitions**, write in the space below **all the trees** characterized by those sequences..

--

pre-order visit: 8 4 8 2 3 4 5 6
in-order visit: 8 4 2 8 4 3 5 6
post-order visit: ? ? ? ? ? ? ? ?

Name: _____ Surname: _____

Exercise 3 (15 points, 20 minutes): the heapsort algorithm sorts a list by first putting all the elements of the list in the heap, and then taking out from the heap the elements one by one, building the output list. Show after each main iteration of the algorithm the status of the data structures involved in sorting the following list using heapsort: 12 5 3 7 12 9 1 4

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)
begin
    if (x == 1) then return 1
    else return f(x/2) + f(x/2)
end
```

Name: _____ Surname: _____

Exercise 5 (15 points, 10 minutes): given the following Burrows-Wheeler transform of a string S, recompute S.

a m m r m \$ e a