

Algorithms and Data Structures, Academic Year 2013/2014

International Bologna Master in Bioinformatics

September 17, 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 extracts the substring S composed by exactly N characters (N-mers), $N < 5000$, with the maximum number of occurrences in a DNA sequence composed by 5000 characters (GCAT). The occurrences of S must be separate, that is, not (even partially) overlapping. The algorithm must provide the following results:

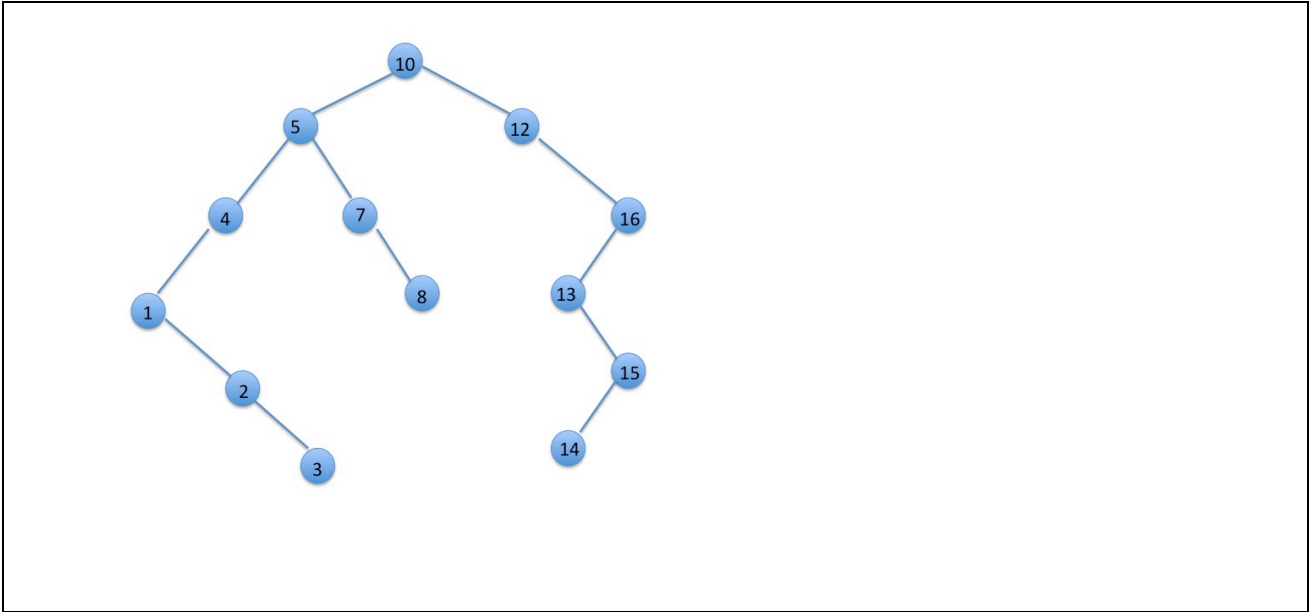
1. How many occurrences of string S appear in the DNA sequence?
2. Which one is the index of the beginning of last occurrence of string S?
3. Which substring is the largest one between two consecutive occurrences of S?

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

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

Name: _____ Surname: _____

Exercise 2 (18 points, 20 minutes): given the following binary search tree, write the ordered set of visited nodes in in-order and post-order, respectively, and re-write it as a balanced AVL tree version and list the sequence of actions made on each node (e.g. LL on node 10, RL on node 7, ecc.) for transforming the binary search tree into a balanced AVL tree.



in-order visit:
- - - - -

post-order visit:
- - - - -

List of actions (LL, LR, RL, RR) on nodes in sequence, to transform the tree into AVL tree.
- - - - -
- - - - -
- - - - -

Name: _____ Surname: _____

Exercise 3 (15 points, 20 minutes): please provide the ordered sequence of visited nodes in a Breadth-First-Search (DFS) visit of the directed graph G , starting from node A , by using the Adjacency List implementation (also show the Adjacency List data structure, and plot the graph).

$G=(V,E)$, $V=\{A, B, C, D, E, F, G, H, I\}$,

$E=\{(A,D)(A,E)(B,C)(B,G)(C,H)(E,B)(D,F)(F,E)(H,I)(I,F)(I,A)\}$

Name: _____ Surname: _____

Exercise 4 (15 points, 10 minutes): Please compute the time and space complexity for the computation of this recursive function and determine the mathematical expression implemented, for any $x > 0$.

```
function f(int x)
begin
    if (x == 0)    then return (-1)
                  else
    if (x <= 2)    then return (f(x-1))
                  else
                  return (f(x-2)-1)
end
```

Name: _____ Surname: _____

Exercise 5 (15 points, 10 minutes): Show the heap structure (by using the array ADT) generated by the insertion of the following list of numbers, and then show the data structure evolution of the sorting process of the same set of numbers by using the quicksort algorithm, and selecting the best pivot at each round:

74 54 27 18 07 33 90 57 43 65

HEAP obtained after insertion of above numbers in sequence:

0	1	2	3	4	5	6	7	8	9	10	11	12	13

QuickSort: (put a circle around the pivot in each line/round)

74 54 27 18 07 33 90 57 43 65

round 1 _____

round 2 _____

... _____
