ALGORITHMS *Degree in Computer Science Engineering*

Course 2016/17

EXAM EXERCISES December, 21th 2016

EXERCISES:

1. Given the weighted and directed graph G = (V,E), with V = {a, b, c, …, h} and E determined by the following edges list:

(a,d)	(a,f)	(d,h)	(d,f)	(h,e)	(h,g)	(e,g)	(f,g)	(f,b)	(f,c)	(b,c)
1	2	1	6	2	7	2	3	4	5	2

a) Draw the graph G.

b) Write the depth-first traversal over G, starting with vertex **a**.

c) Write one topological ordering for the vertices of G.

d) Draw a minimum spanning tree of the undirected underlying graph, indicating its total weight.

e) Draw the tree with the shortest paths from vertex **a** to all other vertices of the graph, indicating for each vertex the minimum distance calculated.

2. Given the weighted and directed graph G = (V,E), with V = {1, 2, 3, ... , 8} and E determined by the following edges list:

(1,2)	(1,3)	(2,7)	(3,2)	(3,4)	(3,5)	(3,7)	(4,5)	(4,8)	(5,6)	(5,8)	(7,5)	(7,6)	(8,6)
5	2	2	3	4	2	1	3	1	2	1	1	4	2

Draw the graph avoiding edges crossing, and then draw:

a) A tree associated to a depth-first traversal starting with vertex **1**.

b) A tree associated to a breath-first traversal starting with vertex **1**.

c) A minimum spanning tree of the undirected underlying graph, indicating its total weight.

d) A tree with the shortest paths from vertex **1** to all other vertices of the graph, indicating for each vertex the minimum distance calculated.

e) A modification of the graph where their vertices are aligned and ordered topologically.

3. Represent by a decorated graph all the game situations that could be reached with a heap of 5 matches for the variant of Nim's game seen at theory class.

4. Represent by a decorated graph all the game situations that could be reached with a heap of 4 matches for the variant of Nim's game seen at theory class. ¿How do you identify the initial situation?

5. Explain how to calculate an array with the topological sort of each node of a directed acyclic graph, traversing all its nodes: which procedures would you use and how would you complete them?, which would be the schema (or pseudocode) of the topological sort function?.

6. Examples of the application of the Divide and Conquer theorem to solve recurrences. Formulate the theorem. Then, build a table with the following columns:

   - In which *case* the theorem is applied (best case, average case, worst case, or all)
   - *Characterisation* of the case (when the indicated case of the previous column is produced?)
   - Recurrence *equation* (of the form $T(n) = l\,T(n/b) + c\,n^k, n > n_0$)
   - Result

7. Compare *Merge sort* with *Quick sort* from the algorithm design point of view (design technique, steps of the used technique, strategies to improve the execution time, …) from its computational complexity (recurrence relations for each case, justified and solved, theorem for solving them, …).

8. Explain the use of an *ad hoc* function to deal with the base case in divide and conquer algorithms, whose schema will be indicated in the answer. How such a function is used in the Quick sort algorithm?.

9. Develop the calculations for the call Fibonacci(4) to explain the differences between the use of a divide and conquer strategy opposite to dynamic programming.

10. Using the dynamic programming technique, design an algorithm to calculate Fibonacci(n) in strict linear time and constant storage space.

11. Calculate a binomial coefficient C(n,k), using the dynamic programming technique in such a way that memory needs are minimum:

    a. Propose a specific example, starting with Pascal's triangle, to explain the algorithm.

    b. Determine its temporal and spatial computational complexity.

12. Compare the following algorithms with the Divide and Conquer function as schema of the design technique, including it in the answer:

    a. Quick sort with random pivot

    b. Binary search