Report for Programming Problem 2

Team:

Student ID: 2019217590 Name Sancho Amaral Simões

Student ID: 2019243695 Name Tiago Filipe Santa Ventura

1. Algorithm description

A *dynamic programming top-down* approach was used. The approach consists in a *DFS* like algorithm, since it traverses all the nodes belonging to the tree. When in a node *i*, the current problem divides into two smaller problems: calculating the minimum vertex cover by using the current node *i* or by not using it. Because of this, some computations may be repeated. Hence, the **memoization** technique used to avoid these redundant calculations: an array serves the purpose of a cache structure and stores all the computations of the solutions performed until the current moment.

2. Data structures

Member – a class that allows the creation of objects that store all the contacts of a member of the pyramid scheme: its recruited and the recruited members.

Solution – a class that allows the creation of objects that contain the information of a solution for a subproblem: the number of members and their total value. This class also provides a method that allows the choice of the best Solution, given two of them. A bidimensional array of objects of this class was created in order to use **memoization**.

3. Correctness

The implemented solution allowed the group to reach the 200 points achievement since it uses an adapted version of the well-known algorithm **minimum vertex cover**, but instead of applied to any graphs or binary trees, it is applied to general trees, since each node can have more than two children. The exact algorithm for the minimum vertex cover for binary trees is well studied (check the bibliographic references) and uses the optimal substructure of the problem: when the current problem is broken down into its two subproblems (use or don't use the current node) it may happen that a solution for one of these subproblems has already been calculated.

4. Algorithm Analysis

Time complexity: $O(N^2) – O(2^N)$ is avoided due to the use of dynamic programming.

Spatial complexity: $O(1)$ – no dynamic structures are used. The size of each array is statically typed and does not resize according to the input.

5. References

- https://www.geeksforgeeks.org/vertex-cover-problem-set-2-dynamic-programming-solution-tree/
- https://arxiv.org/pdf/1705.00216
- https://riptutorial.com/dynamic-programming/example/25808/minimum-vertex-cover
- https://iq.opengenus.org/vertex-cover-problem/