

## ASSIGNMENT 4

Due date: July 24, 2024 16:00 (Waterloo time)

**WARNING: To receive credit for this assignment, you checked “I agree” to the academic integrity declaration. Please keep all the rules in mind as you complete your work.**

**Coverage:** Through Module 10

This assignment consists of a written component and a programming component. Please read the instructions on the reference page for assignments carefully to ensure that you submit each component correctly.

Please check the pinned FAQ in the discussion forum for corrections and clarifications.

Note: In assignment questions that specify the use of a particular paradigm, you are expected to come up with a new algorithm using that paradigm. It is not sufficient to implement a class example as a helper function and declare that the paradigm has been used.

### Written component

For full marks, you are expected to provide a brief justification of any answer you provide.

- W1. [6 marks] In Assignment 2, we considered a greedy algorithm for finding a maximum size cluster in a graph. Here we will consider the algorithm as an approximation algorithm.
- (a) [1 mark] What is the approximation ratio when the instance is a complete graph on at least two vertices? Briefly justify your answer.
  - (b) [1 mark] What is the approximation ratio when the instance is a tree on at least two vertices? Briefly justify your answer.
  - (c) [4 marks] Give a family of graphs for which the approximation ratio cannot be bounded above by a constant. That is, provide a function  $f(n)$ , where  $f(n)$  is not a constant function, such that the ratio between the approximate and the optimal solution is at least  $f(n)$ . You do not need to provide a graph for each value of  $n$ , but you should show how to form an infinite family of graphs of increasing size. As in W2 of Assignment 2, the statement should hold for any ordering of the vertices, no matter how ties are broken.
- W2. [6 marks] In this question, we will consider two possible backtracking algorithms for CLUSTER PARAMETERIZED BY CLUSTER SIZE, defined as follows:

**Input:** A graph  $G$  and a positive integer  $k$

**Output:** Yes or no, answering “Is there a cluster in  $G$  of size  $k$ ?”

**Parameter:**  $k$

For each of the algorithms described below, answer the following two questions:

- Is the worst-case running time of the algorithm in  $f(k) \cdot n^{O(1)}$ ?
- Is the proposed algorithm guaranteed to produce the correct answer?

- (a) [3 marks] Form a search tree in which the root corresponds to the empty set. At each node, form a child for each node not already in the set that is a neighbour of all nodes in the set (vacuously true for the empty set). Stop after  $k + 1$  levels, producing true if a node is formed and false otherwise.
- (b) [3 marks] Order the vertices. Form a search tree in which the root corresponds to the empty set. At each node, for the next vertex in order, create one set that contains the vertex (if it and the current contents of the set form a cluster) and one that does not contain the vertex. Stop after  $k + 1$  levels, producing true if a node is formed and false otherwise.

W3. [9 marks] In this question, you will use hill-climbing for the problem of REPRESENTATIVE SETS, as defined in Assignment 2.

- (a) [3 marks] Describe a hill-climbing approach, including the initial solution and how it is updated at each step. For full marks, be sure to explain why your approach can be classified as hill-climbing.
- (b) [3 marks] Provide an input on which your approach is guaranteed to produce the optimal solution. Your input must contain at least 4 sets, and each set must contain at least 2 values. Briefly justify your answer by explaining how your approach performs on the provided input and why the solution is optimal.
- (c) [3 marks] Provide an input on which your approach is **not** guaranteed to provide the optimal solution. Your input must contain at least 4 sets, and each set must contain at least 2 values. Briefly justify your answer by discussing both the output of the approach and the optimal solution.

W4. [3 marks] An algorithm for REPRESENTATIVE SETS forms a solution  $\mathcal{B}$  as follows: considering each set in  $\mathcal{A}$  one by one, randomly choose whether or not to put the set in  $\mathcal{B}$ . The probability of adding a particular set is  $1/2$ , so the probability of omitting it is also  $1/2$ . For the purpose of this question, we’ll define “high probability” (of producing the correct answer) to be any probability that is greater than  $3/4$ .

- (a) [1 mark] Does the algorithm fit the definition of a Las Vegas algorithm?
- (b) [1 mark] Does the algorithm fit the definition of a Monte Carlo algorithm?
- (c) [1 mark] Is there ever a reason that one might choose to use a randomized algorithm for a problem in  $P$ ? Briefly justify your answer.

## Programming component

Please read the information on assignments and Python carefully to ensure that you are using the correct version of Python and the correct style. For full marks, you are required not only to have a correct solution, but also to adhere to the requirements of the assignment question and the style guide, including aspects of the design recipe.

Although submitting tests is not required, it is highly recommended that you test your code. For each assignment question, create a testing file that imports your submission and tests the code. Do not submit your testing file.

For any of the programming questions in this assignment, you may import any of the following files: `check.py`, `graphs.py`, and `equiv.py`, as well as built-in modules such as `math` and `copy`.

Be sure to read the instructions on the Python requirements page to ensure that you have imported the files as required.

- P1. *[14 marks]* Write a function `cluster_backtrack` that consumes a graph and produces a list of IDs of vertices that forms a cluster of maximum size. Your function should use backtracking.

Submit your work in a file with the name `clusterbacktrack.py`.