

ASSIGNMENT 3

Due date: July 10, 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 7

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. For example, using binary search or mergesort is not sufficient for a problem asking you to use divide-and-conquer.

Written component

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

W1. [10 marks] In this question, we will use dynamic programming for the following computation:

Given inputs x_1, \dots, x_n and y_1, \dots, y_n such that $x_1 < x_2 < \dots < x_n$, we wish to compute $A(1, n)$, where the following definitions hold:

- $A(i, i) = y_i$ for $1 \leq i \leq n$
-

$$A(i, j) = \frac{A(i+1, j) - A(i, j-1)}{x_j - x_i}$$

for $1 \leq i < j \leq n$

- [2 marks] Calculate $A(1, 3)$, where $x_1 = 1$, $x_2 = 2$, $x_3 = 3$, $y_1 = 1$, $y_2 = 4$, and $y_3 = 9$. For full marks, show your work.
- [2 marks] What are the base cases and their values?
- [2 marks] What is the smallest possible shape and size of a single table used to calculate $A(1, n)$?

- (d) [2 marks] Draw the table needed to solve the problem for $A(1, 4)$. Label each entry with the information stored in that location as well as the order in which it is filled. For the ordering, fill in all the base cases with the number 0 and the rest of the entries using numbers 1, 2, and so on to show the order in which entries should be evaluated. Any nonzero number should appear at most once.
- (e) [2 marks] State and briefly justify the running time of the algorithm for $A(1, n)$ in Θ notation as a function of n . You can assume that each mathematical operation can be executed in $\Theta(1)$ time.

W2. [5 marks] In this question, we will consider lower bounds on MAXIMUM SIZE CLUSTER, as defined below:

MAXIMUM SIZE CLUSTER

Input: A graph G

Output: The maximum size of any cluster in G

For your bound, you will consider only algorithms in which the key operation is specifying two vertices and asking whether they are adjacent. Algorithms may not access the graph in any other way.

The lower bound should be as big and as precise as possible; do not use order notation. You should explain an adversary strategy and how the strategy can be used to prove the lower bound.

- (a) [1 mark] Describe an adversary strategy.

Remember that an adversary can store and compute whatever it wishes, without worrying about limits on resources, but is unable to know what the algorithm is going to do next.

- (b) [4 marks] State and prove the adversary lower bound in terms of n .

W3. [8 marks] Using all the steps of the recipe shown in lecture, prove that REPRESENTATIVE SETS DECISION, defined below, is in NP.

REPRESENTATIVE SETS DECISION

Input: A set \mathcal{A} of sets of numbers and a positive integer k

Output: Yes or no, answering “Does there exist a subset \mathcal{B} of \mathcal{A} such that:

- the union of all the sets in \mathcal{B} is equal to the union \mathcal{U} of all the sets in \mathcal{A} , and
- the size of \mathcal{B} is at most k ?”

W4. [7 marks] Using the fact that CLIQUE is NP-complete, follow the steps in the recipe to show that CLUSTER DECISION is NP-complete. You may assume that CLUSTER DECISION is in NP.

CLUSTER DECISION

Input: A graph G with positive weights on edges, a positive integer k , and a positive integer B

Output: Yes or no, answering “Is there a cluster in G of size k and with total value at most B ?”

Since the first two steps of the recipe have been completed, start your work at the third step.

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`, `grids.py`, `graphs.py`, and `equiv.py`, as well as built-in modules such as `math` and `copy`.

P1. [18 marks] In this question, you will implement the dynamic programming algorithm for STRING EDITING, defined as follows:

Input: A source string S , a target string T , and integer costs c_a for addition, c_d for deletion, and c_s for substitution

Output: The lowest cost of any editing sequence that can be used to edit s into t

For example, we can edit the string `cats` into the string `chats` by a single addition (of `h`), into the string `cat` by a single deletion (of `s`), and into the string `cots` by a single substitution (of `a` into `o`). We could also edit `cats` into `cots` by a deletion (of `a`) followed by an addition (of `o`). If $c_a = 1$, $c_d = 2$, and $c_s = 10$, using one deletion and one addition

would have a total cost of $c_d + c_a = 3$, which would be cheaper than a substitution at a cost of $c_s = 10$.

The problem can be solved using dynamic programming, where $C[i, j]$ is defined as the minimum cost to edit $S[:i]$ into $T[:j]$. The following facts can be used:

- If $S[i - 1] = T[j - 1]$, then $C[i, j] = C[i - 1, j - 1]$.
- Otherwise, $C[i, j] = \min\{C[i - 1, j - 1] + c_s, C[i - 1, j] + c_d, C[i, j - 1] + c_a\}$

Write a function `string_edit` that consumes two strings `source` and `target` and integers `add_cost`, `delete_cost`, and `sub_cost` and produces the cheapest cost of an edit sequence from `source` to `target`. **Your function must use dynamic programming.**

For full marks, you must use grids. To use the module `grids.py`, use the line `from grids import *`; do not include the code for `grids.py` directly in the file you submit.

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

- P2. [7 marks] In this question, you will write a verification algorithm for CLUSTER DECISION, as defined in Question W4.

Write a function `cluster_verify` that consumes a graph `graph`, two integers `size` and `bound`, and a `certificate`. You cannot make any assumptions about the type of `certificate`.

Your function should produce `True` if `certificate` is a list of strings of length `size`, where the strings are distinct IDs of vertices in `graph` that form a cluster with total value at most `bound`. Otherwise, your function should produce `False`.

Here are a few examples, assuming that `graph_four` is Sample graph 4. Then, `cluster_verify(graph_four, 2, 17, ["d", "g"])` will produce

`True`, since the two vertices with IDs in the certificate form a cluster of total value 11, and $11 \leq 17$. However, `cluster_verify(graph_four, 3, 17, ["b", "f", "g"])` will produce `False`, since the vertices with the IDs in the certificate do not form a cluster, and `cluster_verify(graph_four, 3, 14, ["e", "f", "g"])` will produce `False`, since the total value of the cluster is 16, which is greater than 14.

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