

CS 323 Homework 7

Due by Sunday, April 2, 2017 8:45 PM

Submission instructions

Submit your assignment through the QTest system, using course ID: **CS323** and exam ID: **hw07**. Write all your code into the code boxes provided in QTest, and make sure that it works correctly by pressing the “Execute” button. If your program is composed of multiple classes, make the first one public and all the other ones not public (just omit the public visibility modifier for all classes except the first one). No email submissions are accepted. No late submissions are accepted. Include a collaboration statement in which you acknowledge any collaboration, help, or resource you used or consulted to complete this assignment. This section must be written even if you worked on the assignment alone.

1 Rod Cutting (55 points)

Implement the “rod cutting” algorithm in three different ways: direct recursive implementation, top-down recursion with memoization, and bottom-up iteration with memoization.

- (15 points)

```
public double rodCut1(int length, List<Double> prices, List<Integer> resultCuts)
```

Finds the most profitable set of cuts for a rod of a given length. The selling prices for each rod size is given in the List `prices`. The List `resultCuts`, initially empty, will be filled by the method with the optimal list of cuts found by the algorithm. The method returns the total profit. The algorithm used is: direct recursive implementation.

- (15 points)

```
public double rodCut2(int length, List<Double> prices, List<Integer> resultCuts)
```

Finds the most profitable set of cuts for a rod of a given length. The selling prices for each rod size is given in the List `prices`. The List `resultCuts`, initially empty, will be filled by the method with the optimal list of cuts found by the algorithm. The method returns the total profit. The algorithm used is: top-down recursion with memoization.

- (15 points)

```
public double rodCut3(int length, List<Double> prices, List<Integer> resultCuts)
```

Finds the most profitable set of cuts for a rod of a given length. The selling prices for each rod size is given in the List `prices`. The List `resultCuts`, initially empty, will be filled by the method with the optimal list of cuts found by the algorithm. The method returns the total profit. The algorithm used is: bottom-up iteration with memoization.

- (10 points) In the main method of your program, write enough test cases to thoroughly test all the methods you implemented. Experiment with different input sizes so you can observe the difference in efficiency of the various algorithms.

2 Minimum Edit Distance (45 points)

Implement the minimum edit distance algorithm using an iterative bottom-up implementation with memoization.

- **(35 points)** `public static int editDistance(String x, String y, int insertCost, int deleteCost, int replaceCost, List<String> resultOperations)`
Calculates and returns the minimum edit distance between strings `x` and `y`, calculated using the provided costs for insert, delete, and replace operations. The List of strings `resultOperations`, initially empty, will be filled by the method with the sequence of operations to transform `x` into `y`. The strings contained in this list have the following format:
 - `insert c` for an insert operation, where `c` is the character being inserted.
 - `delete c` for a delete operation, where `c` is the character being deleted.
 - `replace c1 with c2` for a replace operation, where `c1` is the character being replaced by character `c2`.
- **(10 points)** In the main method of your program, write enough test cases to thoroughly test your method.

Grading criteria

- Programs that do not compile in QTest get zero points.
- -10 points for missing collaboration statement.
- -10 points for missing comments or bad usage of comments.