

CISC/CMPE 471 Project

Due: April 20, 2021 at 11:59 PM EST

1 Coding Grading - 20 points

1.1 Style - 5 points

These will be checked and graded automatically using a variety of tools. They are standard in a industry and research labs to enforce good coding standards being committed to a repository.

- ☐ Every function is at most 30 lines.
- ☐ Every line doesn't exceed 80 characters.
- ☐ Every function is documented with PyDoc style comments.
e.g. <https://stackoverflow.com/questions/34331088/how-to-comment-parameters-for-pydoc>
- ☐ Every Rosalind node should have its own file in the format `ACRN.py` where `ACRN` is the acronym for the node in the tree view.

1.2 Correctness - 5 points

The algorithm will be graded on correctness based on the tests you provide as well as Rosalind datasets. You must provide the Rosalind tests as part of the project and demonstrate that it works on their data. We will also test it independently with the same re-download of the Rosalind data. If the project node you choose doesn't work on the tests, we will test it on the ancestor nodes and grade you based on how many of them are working. You must also provide the test setup for all the ancestor nodes.

1.3 Testing - 5 points

Note: Do these tests as you go and not at the end! They're actually useful for you to debug your code!

- ☐ All your test files should be in the format `*_test.py`, one test file for each node.
- ☐ You must use the Python unit test framework here.
- ☐ Every testing function should have the format `test_function_name_XXX` where `function_name` is the function's name, and `XXX` is the test name for that function. You should have multiple tests per function.

Testing will be graded based on `Radon` for test coverage. The grade (A-F) outputted by `Radon` will be used. For full marks, every function must have full code coverage tests.

1.4 Extension - 5 points

In addition to the previous sections, the extension to the code will have a rubric.

- 5 points - The extension is novel, not marginal, and has applications.
- 3 points - The extension is marginal, but not trivial, and has applications.
- 1 points - The extension is trivial.

1.5 Report Code

You will have code generated for your report figures. For graphs, you should not output an image, but data to be used in LaTeX directly. These should go in a main file `figures.py` and subfiles `figures_figureName.py` where you will have multiple files with different `figureName`.

2 Report Grading - 20 points

- ☐ Your report grade will be based on grammar and spelling, presentation, clarity, and adherence to the instructions. Marks will be deducted for a lack of these items. A handy link for checking clarity is <http://hemingwayapp.com/>.
- ☐ It must be in the ACM LaTeX template located here.
- ☐ Computing classification system keywords are mandatory and should be generated **here** based on your project type and copy/pasted into your report in the right place.
- ☐ Your project should be **eight to ten pages**, in a two column format, submitted as a single file PDF. The sample template to start from is **sample-sigconf.tex**.
- ☐ For figures it's handy to get them to span `\textwidth`. If you want it to span the whole two columns use `\begin{figure*}`.
- ☐ You must refer to figures, tables, etc... As `Figure~\ref{fig:figureName}` in the L^AT_EXcode. This looks like Figure 1 or Table 1 in practice. You must label your figure captions, e.g. `\caption{This is a caption} \label{fig:figureName}`. See [here](#).
- ☐ Your report must have the following sections: Abstract (as `\begin{abstract}`, no more than 250 words), Introduction, Related Work, Approach, Results & Discussion, Conclusion, References.
- ☐ Replace the copyright and acm section with

```
\setcopyright{rightsretained}
\copyrightyear{2021}
\acmYear{2021}
\acmDOI{}
\acmConference[Queen's University - CISC471 2021]
  {Queen's University - CISC471 2021: Computational Biology}
  {April 20, 2021}
  {Kingston, ON, Canada}
\acmBooktitle{}
\acmPrice{}
\acmISBN{}
```

2.1 Grading

The report will be graded on the above checklist, and also the rubric below.

- > 16 points. The work is high quality publishable quality. The supporting graphs, and arguments are clear, to the point, and accurate. The extended algorithm has clearly been shown to be better or not beyond a reasonable doubt. There are no spelling or grammatical errors.
- 14 points. The work is marginally publishable. The report is of good quality, but it is not beyond a reasonable doubt that the algorithm is better or not. There is still some exploration left to do to prove it. The writing could be improved.
- 12 points. The work is not publishable. The report is poorly written. The arguments are not convincing. There is a skeleton of a good argument to be made, but it is very rough.
- < 10 points. The writing is difficult to follow. The graphs are difficult to understand or irrelevant to the argument. There are logical assumptions that are false. There are spelling, and grammatical errors.

3 Report Format

The report must be between 8 and 10 pages as before. Here is a breakdown of the sections with approximate page counts.

3.1 Abstract

At most 250 words. Describe the proposed algorithm, and how it fits into the space of the problem.

3.2 Introduction

Around 1 page. Describe the problem, and your proposed algorithm. Why is it better than other current methods?

3.3 Related Work

Around 1 page. A summary of related work in the area. At least 10 citations will be required.

3.4 Approach

Around 2-4 pages. Summarize your approach. What is your extension? What assumptions do you make to improve the performance?

3.5 Results & Discussion

Around 2-4 pages. Summarize results with graphs, tables, figures, etc... here. Discuss their meaning and convince the reader that your approach is either better, or didn't work out. Try to understand why this is the case.

3.6 Conclusion

Around 1/2 page. Summarize the whole paper in at most a few paragraphs. Some readers only read the Introduction/Conclusion, so make it so just reading those two paint a picture to entice the reader to read the rest of the paper.

3.7 References

At most 1 page. This section should be auto-generated by L^AT_EX.

4 Oral Examination - 10 points

You will be individually asked a set of questions based on the code you wrote, as well as the report. These will be made by appointment. **It is everyone's duty in the project to fully understand the algorithm that is being implemented, the extension, and the literature in the field that you cite.** You will be asked questions like:

- What does this variable x do? What happens if we change it to go up or down?
- Overall, what was your algorithm? Describe it? Why did you choose to do it this way? What happens if you did it this way?
- What have other methods done in the field? Why is yours different?

5 Suggested Milestones

- April 2 - Finish all Rosalind coding
- April 9 - Have initial results for algorithm extension, prepare all report except for approach, results, discussion as they may change with more fine tuning.
- April 16 - Iterate on graphs, approach, results until have something that works decently.
- April 20 - Prepare and submit both coding and report.