ENGSOC
ONTARIO TECH ENGINEERING STUDENTS' SOCIETY

Internal Engineering Competition 2022

PROGRAMMING | Competitor's Package

November 5th, 2022

Ontario Tech University

Table of Contents

Introduction

Programming Competition Contacts

Ontario Tech EngSoc VP Internal & IEC Chair, Natasha Naorem internal.engsoc@ontariotechu.ca natasha.naorem@ontariotechu.net
Ontario Tech EngSoc VP Academics, Daniel Jeon academic.engsoc@ontariotechu.ca daniel.jeon@ontariotechu.net

Rules

1) You can use the internet but copying code will result in penalties.
2) Any and all publicly used resources should be given proper attribution.
3) You can use any language, package, library in order to complete the challenge.

Competition Description

Problem Background

The objective of this challenge is to find the optimal path of a mobile hospital visiting villages in order to maximize the number of patients treated given a limited travel range. The second objective of the challenge is to display the path visually using a GUI.

- The mobile hospital has a limited travel range, this range should be easily adjustable. Assume a range of 1000 units for all initially provided test cases.
- The mobile hospital can start at any chosen village.
- Each village has a specified number of patients that need treatment.
- If the mobile hospital reaches a resupply depot the travel range is replenished back to its original range. The travel range of the mobile hospital cannot surpass the original range at any given time.
- Coordinates are on a flat 2D cartesian plane. Distance is measured using euclidean distance.
- 3 test cases will be provided mapping the village and resupply nodes.
- At your final presentation we will ask you to run unknown hidden test cases.
- The GUI should be functional and show the direction of travel.

The quality of result (QoR) will be calculated as:

QoR = Number patients treated

Competition Deliverables

Expected Output Files - .csv

- Naming convention: `team_team_number_original_map_name.csv`.

Report/README (Documentation of Code & GUI)

- Keep the writing short, maximum 2 pages of written content.
- Outlines the engineering process followed to get the solution.
- Discuss the solution.
- Discuss possible improvements to your design.
- Document images relating to your GUI.

Presentation

- Maximum 10 minutes of presentation allowed, you will be provided an additional 5 minutes of Q&A.
- Summarize report content.
- Demonstrate solution of given test cases.
- Demonstrate solutions of hidden test cases which will be provided at the presentation.
- All team members should be present.

Assessment and Judging

- Avoid reading off of a script and do your best to simulate a professional context.
- Use some form of presentation media (Microsoft PowerPoint, Google Slides, Canva, etc)
- All team members should contribute to the presentation
- Aim to highlight information already covered in your report and refrain from introducing new information

Submission & Deadline

- All relevant material including: solution code, example result, report and presentation must be submitted by the end of the competition time.
- **Code submissions will only be accepted via GitHub: github.com/danielljeon/IEC2022.**
    - Please upload your submission to your specified group number's branch, failure to properly upload your code and solutions will result in a penalty!
- **Presentation & Report will only be accepted via Email: internal.engsoc@ontariotechu.ca.**
- The files must be submitted before 3pm (subject to change) on Saturday November 5th. Late entries will result in a 50% reduction in the marking of that deliverable.

Rubric

Code Performance and Result:	**50%**
Solution is valid, accurate and formatted correctly	10%
Solution QoR *(Each team's QoR may be evaluated against other teams' QoR)*	10%
Solution QoR in relation to performance *(Performance may be evaluated by, but is not limited to: time, complexity and resource utilization)*	5%
GUI is accurate and functionally useful	10%
Repeatable results for hidden test cases	15%
Overall Strategy & Professionalism	**20%**
Code documentation	5%
Code simplicity and complexity	5%
Professional code styling	5%
Appropriate use and organization of packages, libraries, etc	5%
Presentation:	**30%**
Design Process	5%
Design justification and defence	5%
Potential future points of design improvement	5%
Presentation Quality & Visual Aids	2%
Voice and articulation	3%
Presentation flow and timing	2%
Team participation	5%
Responses to Questions	3%
Penalty: Plagiarism	**-50%**
Penalty: Documents submitted late or improperly	**-50%**
Penalty: Absent Team Member	**-25%**