Artificial Intelligence Homework Assignment

2^{nd} year, 2^{nd} semester, 2019-2020

Costin Badică, Alexandra Albişi, Mihaela Colhon,
Ionuţ Murareţu

Abstract

This document introduces the goals and methodology for developing the Artificial Intelligence homework. The document contains also a description of the homework deliverables. The document targets 2^{nd} year, 2^{nd} semester students of the Computers specialty.

1 Introduction

The goal of your homework is to develop a software for experimenting with heuristic search algorithms for problem solving. The homework is focused on the development of skills for programming search algorithms, covering coding, design, documentation and presentation of results. At the end of your homework you must produce a set of deliverables including: a technical report, source code and experimental data. The problem statement is formulated in Section 8.

This document is typeset using LaTeX [1].

2 Deliverables

You have to produce three types of deliverables for this homework:

i) **Technical report.** It should describe briefly, concisely and clearly your work and achievements for the homework. The description must contain your homework task (problem), how did you develop the software and the outcome of your work. More details are given in Section 3.

ii) **Source code.** The source code should contain the source modules of your software. The software should be developed using a programming language of your choice. More details are given in Section 4.

iii) **Experimental data.** You should test your software using non-trivial input data. You should provide non-trivial input data sets, and the corresponding outputs produced by your software. The source code for generating non-trivial input data will be also provided, together with the description of the methods and algorithms that you have used to generate

the input data. This description will be included in the technical report. More details are given in Section 5.

You have to deliver an archived directory CA_[full_name].zip (e.g. CA_muraretu_ionut.zip, only ZIP archives will be accepted) containing the following two sub-directories:

a. Code. This directory should contain your code and test data.

b. Documentation. This directory will contain all the files related to technical report. This will be written in LaTeX .

If you will not provide the requested file formats, your homework will not be evaluated.

Very important:

- The assignment is individual and should not be copied from the Internet or from other colleagues.
- Multiple files uploaded on Google Classroom will not be taken into consideration.
- You will have 0 mark if you will not submit the assignment.
- Do not miss the deadline. You may not be able to submit the assignment after.

3 Technical report

The technical report must be typeset using LaTeX and provided in electronic form (PDF and sources). For documentation about LaTeX you can consult reference [3].

The technical report must be divided into a number of sections, including:

i) **Cover page.** This is a one-page section containing the title, the name of the student, the group, year and section where the student is enrolled.

ii) **Problem statement.** This section is the introduction of your technical report. It should describe clearly the task of your homework.

iii) **Pseudocode of algorithms** (other than those included in the homework statement). This section must contain the pseudocode description of your algorithms, not included in the homework statement. The pseudocode should use the format introduced in [2]. An example is shown in Figure 1.

iv) **Application outline.** This section must contain a detailed description of the application design containing the following items:

   – The high level architectural overview of the application

```
function SIMPLE-PROBLEM-SOLVING-AGENT(percept) returns an action
  persistent: seq, an action sequence, initially empty
              state, some description of the current world state
              goal, a goal, initially null
              problem, a problem formulation
  state ← UPDATE-STATE(state,percept)
  if seq is empty then
    goal ← FORMULATE-GOAL(state)
    problem ← FORMULATE-PROBLEM(state,goal)
    seq ← SEARCH(problem)
    if seq = failure then return a null action
  action ← FIRST(seq)
  seq ← REST(seq)
  return action
```

Figure 1: Example for typesetting an algorithm.

- The specification of the input data format
- The specification of the output data format
- The list of all the modules in the application and their description
- The list of all the functions in the application, grouped by modules; for every function the following details must be provided:
  - The description of what the function does
  - The description of each parameter, and
  - The meaning of the return value

Your application design must also address the method for automated generation of input test data.

v) **Conclusions.** This section must contain your own conclusions after performing the homework work. Some suggestions about what to include in the list of conclusions are: a summary of your achievements, which were the most challenging and interesting parts and why, future directions for extending the homework in short and long term, a.o.

vi) **References.** You should include here a list of bibliographic references that are cited in the text of the technical report. They can include: textbooks, journal articles, conference proceedings papers, and Web references.

4 Source code

You are free to choose a programming language for developing your homework. Some suggestions are: C, C++, Java, Prolog, and Python, so you are free to

choose the programming language that best fits your preference. The choice of the programming language must be explicitly motivated in the technical report.

The source code must contain the complete sources of your application, including source files, header files, as well as makefiles or other homework configuration files. The code must compile and build using open tools and must not contain any dependencies of integrated development environments.

The following minimum quality requirements must be met by the source code. They will directly influence the grading of your homework.

- You must provide clear instructions for compiling the code and building your application binaries from the source code.
- The code must be well-organized into modules with clear interfaces.
- The code must be well typeset with indentation and clear naming convention of variables, functions, classes and packages.
- The code should be well commented.

5 Experiments and results

In this section you must explain the method that you used for testing your application, as well as the experimental results.

The experimental results will contain:

- You should provide at least 10 non-trivial input data sets of various sizes (small, medium, large and very large), and the corresponding outputs produced by your software.
- A description of the output data that you obtained by running your algorithm, as well as the method that you used to test that this output is correct according to the algorithm specification.
- The execution time of your algorithm, for each input data set.

You must also present the results (including the recorded execution time) of your experiments in a meaningful way for the reader. The presentation method is left for your choice.

6 Schedule

The **hard deadline** for the delivery of your homework is **Friday, May 22, 2020**, end of the day. You must deliver a **single ZIP archive** including all the required homework deliverables.

7 Grading

The grading of your homework will take into account all the elements presented in this document:

- The structure and content of the technical report should follow the requirements stated in Section 3.
- The structure and content of the source code should follow the requirements stated in Section 4.
- The content of the experimental data and results should comply with the requirements stated in Section 5.

8 Homework statement

P1 Suppose two friends live in different cities on a map, such as the Romania map shown in Figure 2. On every turn, we can simultaneously move each friend to a neighboring city on the map. The amount of time needed to move from city i to neighbor j is equal to the road distance d(i, j) between the cities, but on each turn the friend that arrives first must wait until the other one arrives (and calls the first on his/her cell phone) before the next turn can begin. We want the two friends to meet as quickly as possible.

a. Write a detailed formulation for this search problem.

b. Identify a search algorithm for this task and explain your choice.

P2 The Knuth Sequence - start with a number, apply a sequence of factorial, square root, and floor operations, and arrive at any desired positive integer. Eg.: $5 = \lfloor \sqrt{\sqrt{\sqrt{\sqrt{\sqrt{4!}}}}} \rfloor$.

a. Write a detailed formulation for this search problem.

b. Identify a suitable search algorithm for this task and explain your choice.

P3 A basic wooden railway set contains the pieces shown in Figure 3 The task is to connect these pieces into a railway that has no overlapping tracks and no loose ends where a train could run off onto the floor.

a. Suppose that the pieces fit together exactly with no slack. Give a precise formulation of the task as a search problem.

b. Identify a suitable search algorithm for this task and explain your choice.

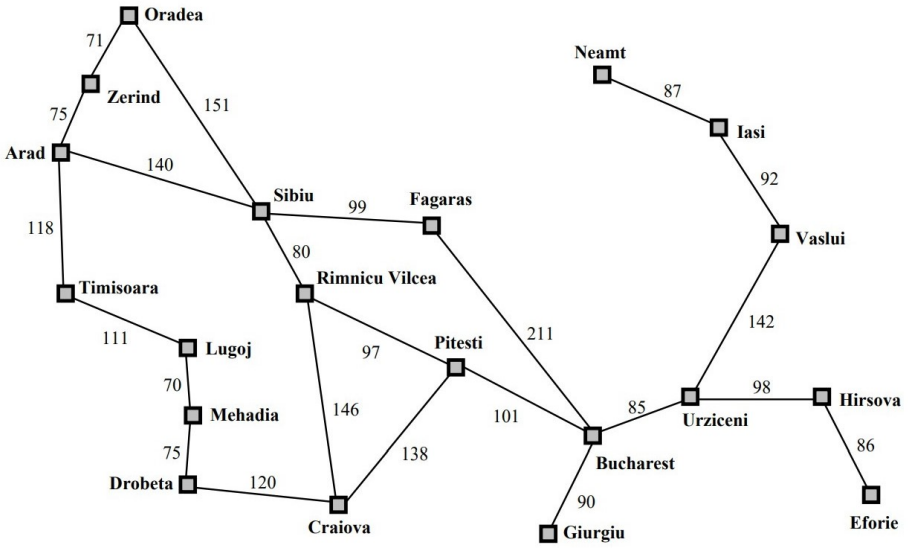

Figure 2: Romania Map

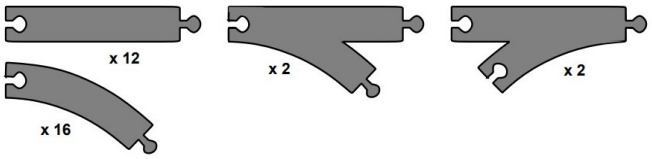

Figure 3: Railway pieces

P4 n vehicles occupy squares (1, 1) through (n, 1) (i.e., the bottom row) of an $n \times n$ n grid. The vehicles must be moved to the top row but in reverse order; so the vehicle i that starts in $(i, 1)$ must end up in $(n-i+1, n)$. On each time step, every one of the n vehicles can move one square up, down, left, or right, or stay put; but if a vehicle stays put, one other adjacent vehicle (but not more than one) can hop over it. Two vehicles cannot occupy the same square.

a. Write a detailed formulation for this search problem.

b. Identify a suitable search algorithm for this task and explain your choice.

P5 Given $n \times n$ grid as in 4, and a configuration of walls you have to place k archers so that they cannot shoot each other. An archer can shoot up, down, left, right and also diagonally an can reach the grid's edges.

a. Write a detailed formulation for this search problem.

b. Identify a search algorithm for this task and explain your choice.

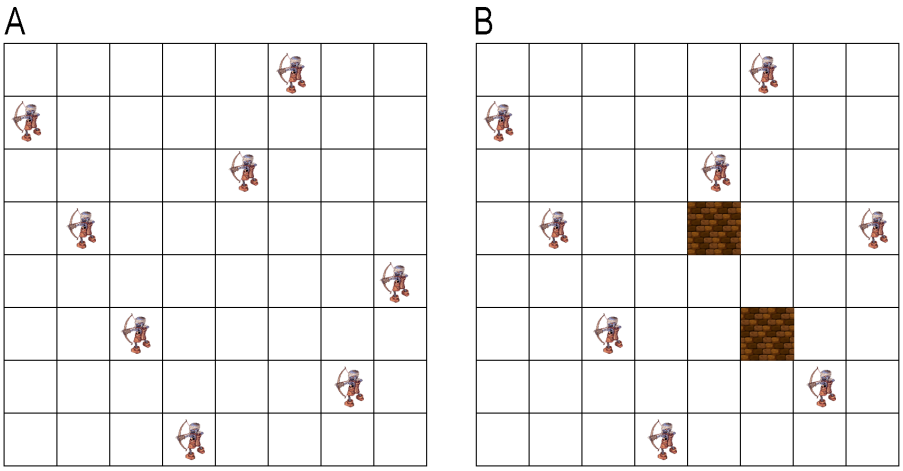

Figure 4: Valid arrangements of archers so that they cannot shoot each other. (A) no walls added. (B) two walls added such that the archer in the last column cannot shoot the archer in the second or fourth column.

8.1 Requirements

Your task is to analyze the problems and fulfill the following requirements:

R1 **Your problem will be assigned using a hash function: 1 + (sum of ASCII codes of your full name [uppercase, no accents and dash, trim] modulo 5)**. Check the list in the assignment file on Google Classroom to see your assigned problem. The assigned problem cannot be changed.

R2 Implement the appropriate code to solve the assigned problem as a search problem.

R3 Present and comment your experimental results and choices in a meaningful way.

References

[1] Leslie Lamport, *LaTeX: A Document Preparation System*. Addison Wesley, Massachusetts, 2nd Edition, 1994.

[2] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3rd Edition, 2010.

[3] LaTeXproject site, http://latex-project.org/, accessed in April 2013.