

CSc74011 Programming project

Goal

Write a general-purpose *modular* search engine that, given a problem and a strategy, returns a solution or failure and uses duplicate node detection (patterned after AIMA's GraphSearch or the class slides).

Demonstrate its performance on multiple problems with multiple strategies.

Stage 1

Write classes for nodes (including state, parent, action that generated it, depth, and cost to node so far, estimated distance to the goal), for problems (including name, start node, goal test, and actions), for the candidates,

Represent and implement at least 3 toy problems and **one challenging problem**.

Show that your code can generate successor nodes by applying each action to a copy of the start state.

Stage 2

Code the following search strategies:

- *At least 2 of* depth-first search with depth cutoff, breadth-first search, iterative depth-first search, and uniform cost search
- *At least 3 of* greedy search, A*, IDA*, RBFS, SMA*, and RTA* (extra credit for more than three)

Collect and output statistics on the number of nodes searched, the maximum size of the candidate set, the depth of the solution, the maximum depth searched, and the elapsed time. **Note:** If you have generalized your search code properly, each of these strategies should simply have its own versions of *Remove*, *Insert* and *Improve*.

Stage 3 (optional)

Here's your chance to look brilliant or get extra credit. Do any of the following, gathering appropriate statistics so you can compare your time/space/success:

- Code a CSP solver that works on the problems you selected.
- Code a GA that works on the problems you selected.
- Meta-reason about what approach to take to a problem (i.e., choose a solver based on the problem characteristics).
- Something else you are eager to try that I approve by March 31.

Submit electronically to susan.epstein@hunter.cuny.edu

DEADLINE #1: Get approval by email for the challenging problem you propose to address and any Stage 3 approach you want to try.

DEADLINE #2: Send **both the code and the output for Stage 1** before class on the date specified on the class website.

DEADLINE #3: Send **both the code and the output for Stage 2** before class on the date specified on the class website.

Final project deadline: Send all of the following **before class on the date specified on the class website.**

Be sure to conform to the posted programming standards.

How to structure your final project submission

Submit separate files for each of the following, [with these labels](#):

- **Code** (heavily documented and in accordance with the posted programming standards)
- **Output** contains **files whose names make clear what they contain** (e.g., breadth-first 8-puzzle runs). Your output must demonstrate how each problem ran and what it calculated. (If you direct output to files, include the resultant files; otherwise capture the screen.)
- **ReadMe** (10 page minimum, the last 20 points on the grading sheet) that includes: what you did, why you did it, what your results are (good to repeat some of the output from the runs here) and clear directions on how I could run the code myself to repeat your results.

[Place all files in one folder that is named YourName](#) (e.g., mine would be SusanEpstein)

Compress the file

Attach it to an email message and send it to susan.epstein@hunter.cuny.edu

Watch your own mail for a “received” message from me.

The idea here is to make it easy for me to fill out that grading sheet form. Don't make me hunt for your answers please.