Sorting as Searching

Part-A

In sorting, given a set of items, the objective is to arrange them in a particular order. In the case of numbers, we generally sort them in ascending or descending order. In this assignment, you need to sort a given set of numbers in ascending order using a search algorithm.

State: A comma-separated list of numbers
Successor function (Action): Swap any two neighboring numbers
Cost: 1 unit per action
Goal test/state: A comma-separated list of numbers sorted in ascending order

For each action, your code should print the current state. The input to the code will be a number ‘n’ followed by ‘n’ numbers. The final output of the code will be the path taken from the start state to the goal state in case of uninformed and informed search methods, and the final state in case of local (hill-climbing) search method.

Example input:
4 4 6.3 9 -3

For the above input, the start state will be {4,6.3,9,-3} and the goal state will be {-3,4,6.3,9}.

Implement the following search algorithms for this:
(1) BFS
(2) DFS
(3) Iterative deepening
(4) UCS
(5) Greedy search using an appropriate heuristic (approximation) function of your choice
(6) A* search using the same heuristic function as used in (4)
(7) Hill-climbing search using an appropriate heuristic (minimizing) function of your choice

Note: The interpretation of the heuristic function used in 4 and 5 is different from that in 6.

Part-B

Run each of the above algorithms at least 20 times for different values of n in {3,4,5,6} by randomly generating different inputs (start states) with `n’ numbers, and calculate the average number of nodes explored by each algorithm for each value of n.

Deliverables

(1) A folder containing your codes and a detailed readme file. You may use any programming language.
(2) A report (PDF) describing the experimental details, results, observations, analyses, etc.

(3) Create a single zipped file name <RollNo_Assig1.zip> containing the above two and upload.

General instructions

(1) Do not paste your codes in the report.
(2) Cite all the resources in the report.
(3) If anything is missing or not clear from the above description, you may make appropriate assumptions and clearly mention them in the report.
(4) A submission which does follow any of the guidelines will be awarded a penalty.
(5) Plagiarism will result in a zero in this assignment, and an additional penalty in the total score in the course.

Penalty for late submission

There will be a penalty of 20% per day. A submission which is >=5 days late will not be evaluated. The time recorded in google-classroom will be considered.