KFUPM

Industrial and Systems Engineering Department

ISE 571 Heuristic Search Methods

Assignment-1 Introduction, Graph Search & Simulated Annealing

Due date: February 15, 2026 – Midnight

Name #	Mohammad Mynul Islam Mahin
KFUPM ID	202528630

What to submit: a PDF report that contains:

- The solutions for the written exercises (Part-I, Exercises #1, 2, 3 & 4), typed or neatly handwritten.
- A link to Google Colab or GitHub for Juypter Notebook of the programming exercise (Part-II).
- Upload this PDF file through Blackboard. Don't email the file to the instructor.

General Instructions:

- Due date is Sunday February 15, 2026 – Midnight
- Anything handed in after the due date will be penalized by 50% for each 24 hours of lateness
- The assignment exercises are scored out of 50 marks, but the final score will be scaled to a total of 5 marks.

Part-I Written Exercises [25 Points]

1. **[5 Points]** MCQs and T/F: Choose the correct answer for each of the following questions.

1.1. Big O specifically describes the limiting behavior of a function (worst-case scenario) when the argument tends towards a particular value or infinity, usually in terms of simpler functions. What is the big-O of this expression: n^3+nlog(n)+log(2n)

a. O(nlogn)

[✓ b. O(n^3)]

c. O(logn)

d. O(n^2)

e. None of the above

1.2. Which blind search algorithm implements FIFO operation for searching the states?

[✓ a. Breadth-first Search (BFS)]

b. Uniform-cost Search (UCS)

c. Bidirectional Search (BS)

d. Depth-first search (DFS)

e. None of the mentioned

1.3. In design problems or strategic functions, optimality is usually traded in for speed gains.

a. True

[✓ b. False]

1.4 A beam search with beam width equals number of nodes in each level is the same:

a. Depth-first search

[✓ b. Breadth-first search]

c. Hill climbing

d. Best-first search

e. Dijkstra's algorithm

1.5 Which of the following statements about the A* algorithm is correct?

a. A* uses only the heuristic function to guide the search.

b. A* algorithm performs an uninformed search like breadth-first search.

[✓ c. A* guarantees the shortest path in a weighted graph, provided the heuristic is admissible and consistent.]

d. A* is a greedy search algorithm that only focuses on the current state without considering future states.

e. A* cannot be used for pathfinding in dynamic environments.

f. None of the above.

2. **[5 Points]** The knapsack problem is a widely studied problem that appears in real-world decision-making processes in a wide variety of fields, such as finding the least wasteful way to cut raw materials, seating contest of investments and portfolios, seating contest of assets for asset-backed securitization, and knapsack cryptosystems. Consider solving this problem using SA. Assume an instance of the problem where we have Knapsack capacity=50 and a set of 7 objects with weights and utilities shown in the following table:

Item	1	2	3	4	5	6	7
Weight	10	12	15	27	30	20	7
Utility	2	1	3	4	1	2	1

Answer the following questions:

a) Define a suitable representation for the problem solution. **[1 Point]**

b) Suggest a suitable operator that can be used to generate the neighborhood of the current solution. **[1 Point]**

c) Define the objective function for calculating the cost of a solution as well as the underlying constraints. Any overload of the knapsack capacity must be penalized with an amount of 50. **[1 Point]**

d) Apply simulated annealing for ONE iteration using an initial temperature of 500, a final temperature of 50 and a geometric decrement rate of 0.85. **[2 Points]**

Solution:

a) Given, total number of objects $= 7$. ∴ A binary vector of length 7 can be used for suitable representation of the knapsack problem: $x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ *and* $x_i \in \{0,1\}$, where i is a particular object. If $x_i = 1$, the item i is selected. If $x_i = 0$, the item i is not selected.

b) To generate neighbors in simulated annealing, bit flip operator can be used as an operator to change the current solution. If $x_i = 1$, using bit flip operator makes it 0. If $x_i = 0$, using bit flip operator makes it 1. ∴ $x'_k = 1 - x_k$, where $k \in \{1, \dots, 7\}$.

c) Let, w_i be the weight of a particular object and u_i be the utility of a particular object. $w = [10, 12, 15, 27, 30, 20, 7]$ and $u = [2, 1, 3, 4, 1, 2, 1]$. ∴ Total weight, $W(x) = \sum_{i=1}^{7} w_i x_i = 10x_1 + 12x_2 + 15x_3 + 27x_4 + 30x_5 + 20x_6 + 7x_7$, and total utility, $U(x) = \sum_{i=1}^{7} u_i x_i = 2x_1 + x_2 + 3x_3 + 4x_4 + x_5 + 2x_6 + x_7$. Given, the knapsack capacity $= 50$. ∴ $W(x) \leq 50$. Maximizing utility is converted into a minimization cost by using negative utility. This is done as simulated annealing often deals with minimization. Also, any overload must be penalized by 50. ∴ f(x) $= -U(x) + 50.I(W(x) > 50)$, where $I(.)$ is an indicator. If overweight, $I(W(x) > 50) = 1$. Otherwise, $I(W(x) > 50) = 0$.

d) Given, initial temperature $(T_0) = 500$, final temperature (T_f) = 50, cooling factor (α) = 0.85. When $T \leq T_f$, stop. Geometric cooling for updating temperature: $T_{new} = \alpha T_0$. Acceptance rule for minimization: for $\Delta f = f(\mathrm{x}') - f(\mathrm{x})$, accept if $\Delta f \leq 0$. Otherwise, accept with probability, $p = e^{-\Delta f / T_0}$. And if random $r \in [0,1]$ satisfies $r < p$, accept. Let the starting point be the empty knapsack. That is, x = (0,0,0,0,0,0,0). ∴ $W(x) = 0$, $U(x) = 0$. It is not overweight. Therefore, no penalty. So, $f(\mathrm{x}) = -0 + 0 = 0$. Temperature at this iteration, $T_0 = 500$. Let, item 4 is flipped. ∴ $\mathrm{x}' = (0,0,0,1,0,0,0)$ and $W(\mathrm{x}') = 27$, $U(\mathrm{x}') = 0$. This is feasible because $27 \leq 50$. So, there is no penalty. So, $f(\mathrm{x}') = -4 + 0 = -4$. Now, $\Delta f = f(\mathrm{x}') - f(\mathrm{x}) = (-4) - 0 = -4$. Since $\Delta f \leq 0$, the neighbor is better. So, it is accepted. Therefore, new solution: $\mathrm{x}' = (0,0,0,1,0,0,0)$ and $T_{new} = 0.85 \times 500 = 425$. Since $425 > 50$, there will be further iterations.

3. **[5 Points]** Consider the following simplified map shown in Figure C.2, where edges are labeled with actual distances between the cities. State the path to go from city A to city M produced by BFS and the path produced by DFS.

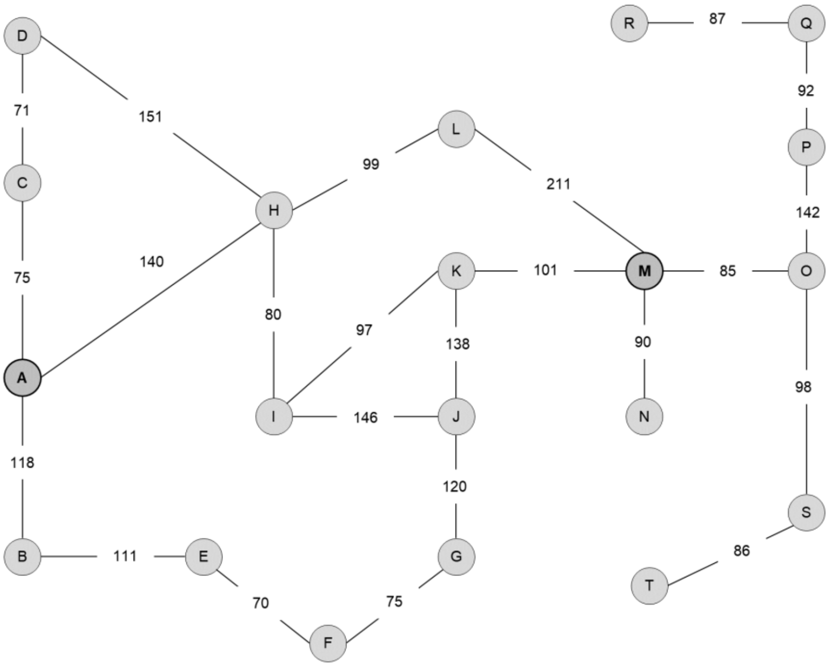
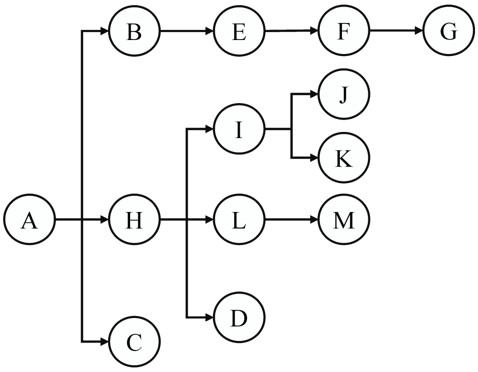


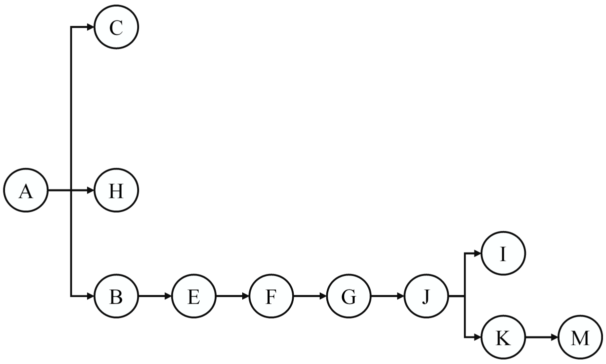
Figure 1 Simplified map.

Answer:

BFS:



DFS:



Route:

A→B→H→C→E→I→L→D→F→J→K→M

Final route:

A→H→L→M

Total distance:

140+99+211=450.

Route:

A→B→H→C→E→F→G→J→K→I→M

Final route:

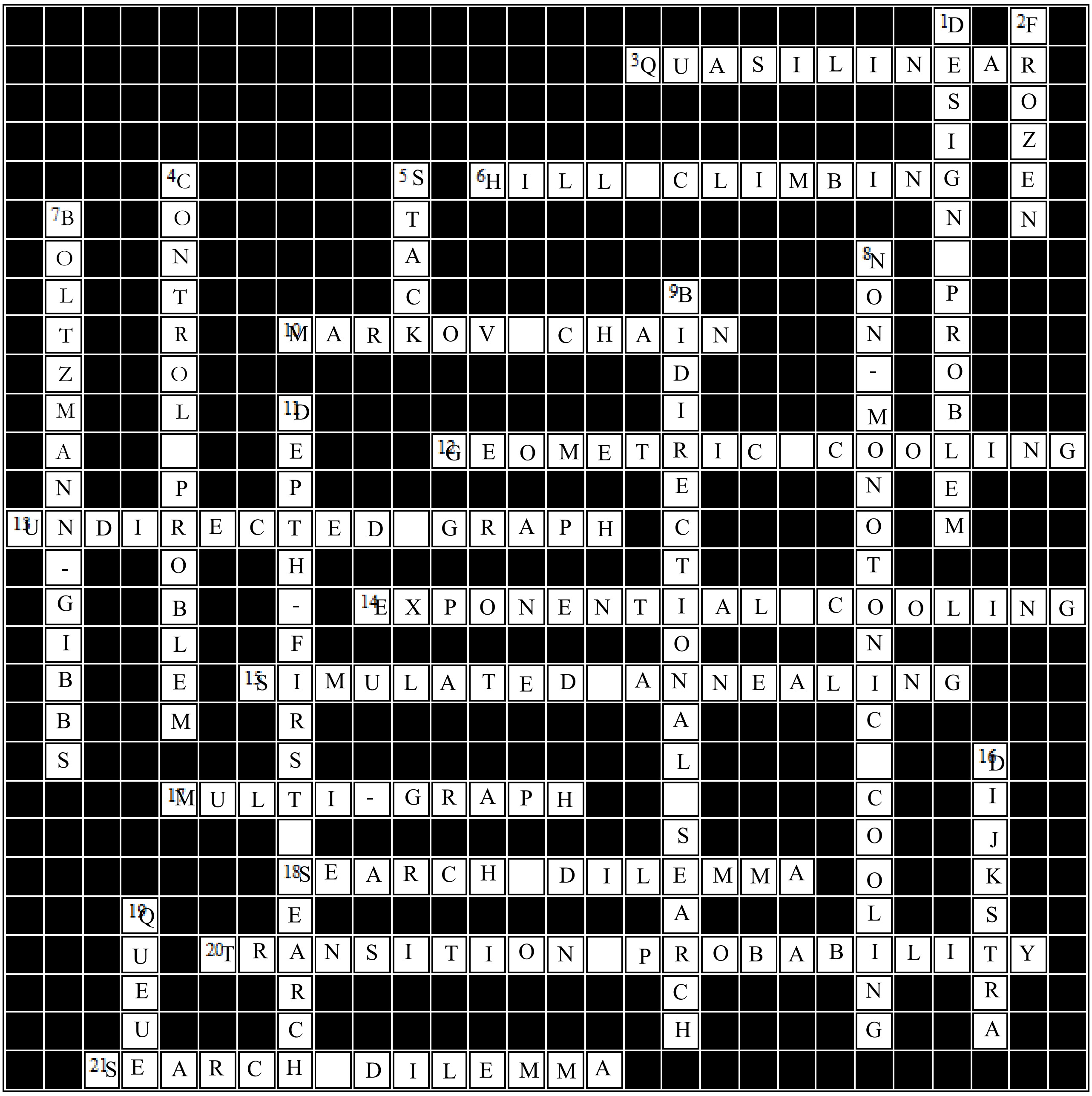
A→B→E→F→G→J→K→M

Total distance:

118+111+70+75+120+138+101=733.

4. **[10 Points]** Crossword puzzle

Solve the following puzzle:



Across

3. a function that is slightly faster than linear complexity.
6. compared to this search algorithm, the main difference is that SA probabilistically allows downward steps controlled by current temperature and how bad move is.
10. a stochastic or probabilistic model that describes a sequence of possible moves in which the probability of each move depends only on the state attained in the previous move.

Down

1. an optimization problem that is used when time is not as important as the quality of the solution and users are willing to wait (sometimes even a few days) to get optimal solutions.
2. state of the system is at which no better or worse moves are being accepted.
4. an optimization problem that needs to be solved repetitively and quickly, in a time span from a few milliseconds to a few seconds.

12. a cooling schedule that decreases the temperature by a cooling factor.
13. type of graph used in LinkedIn to represent users, groups, unregistered persons, posts, skills, and jobs.
14. a cooling process by which the temperature decreases very quickly during the first iterations, but the speed of the exponential decay is slowed down later.
15. an optimization process based on the physical annealing process.
17. a graph in which multiple edges may connect the same pair of vertices.
18. represents the trade-off between exploring new unvisited states or solutions in the search space and exploiting the elite solutions found in a certain neighborhood in the search space.
20. the probability of acceptance or rejection of neighboring solutions
21. represents the trade-off between exploring new unvisited states or solutions in the search space and exploiting the elite solutions found in a certain neighborhood in the search space.

5. a Last-In-First-Out (LIFO) data structure.
7. probability distribution used in transition probability of simulated annealing.
8. a cooling schedule that explicitly takes into consideration how the search is progressing.
9. a search algorithm that combines forward and backward search.
11. a graph traversal algorithm that first explores nodes going through one adjacent of the root, then next adjacent until it finds a solution or until it reaches a dead end.
16. a blind search algorithm that solves the single-source shortest path problem for a weighted graph with non-negative edge costs.
19. a First-In-First-Out (FIFO) data structure.

Hint: Spaces and dashes MUST be used if the answer consists of two or more words.

Part-II Programming Exercises [25 Marks]

Geospatial data is any data related to or containing information about a specific location on the Earth's surface. Answer the following questions:

1. **[5 Marks]** Pick any dataset that contains spatial data and information (preferably related to your course project). Have a look at the list of publicly available spatial datasets here. You can also use Google Dataset Engine or HuggingFace to search for data. If needed, use geocoders to find the geographic coordinates (lat and log) given the location address.

2. **[5 Marks]** Data visualization is a powerful tool to bring data to life, convey information and help in decision making. Spatial data can be visualized in different formats such as chloropleth map, cartogram map, bubble map, hexagonal binning, heat map, cluster map. Write Python script in a Jupyter notebook format to visualize this data in different formats and report the insights that you may observe from these visualizations. Observations can be reported in a markdown cell to be part of the same Jupyter notebook. Here is a sample Jupyter notebook.

3. **[5 Marks]** Pick any two points of interest (preferably in Saudi Arabia) and add markers on the map to highlight these two points.

4. **[10 Marks]** Find and render the routes between the two selected points using BFS, DFS, Dijkstra and simulated annealing. Compare these routing algorithms in terms of time and cost (route length in meters).

Solution

The answers to the above questions can be found in the following GitHub repository: https://github.com/MynulVG/ISE_571_Heuristic-_Search_Methods_g202528630_Assignment_1