CS 470 Homework
Deterministic Search
110 Points

---

1. **Blind search on a grid [24 points]**

   The state space for a search problem is given in Figure 1. The remainder of the search problem is defined as follows:

   - **Initial state**: State a
   - **Goal state**: State o
   - **Actions**: The actions available in any state s are to travel to another state that is connected to s by an edge. The resulting states are generated and added to the `FRINGE` in **alphabetical order**.
   - **Action cost**: The action costs are all 1

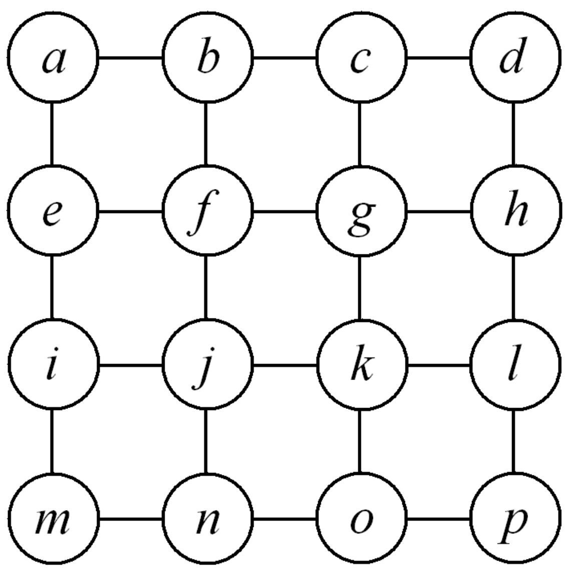


Figure 1: Grid state space for question 5

   For each of the following blind search algorithms:

   - List the states in the order they are **expanded** by the search algorithm
   - Give the path from the initial state to the goal state that is returned by the algorithm.

   Assume that the states that already appear in the `FRINGE` or have already been expanded are not added again. Also, remember to only apply the goal test to a state when it is expanded.

(a) [**8 points**] Breadth-first search

(b) [**8 points**] Depth-first search

(c) [**8 points**] Bidirectional search – Use breadth first starting from both the start and the goal states, expanding one state at a time in each search, alternating searches after each expansion. Begin with the start state. When a node is going to be expanded by one search, check to see if a node with the same state is already in the `FRINGE` of the other search. If it is, then the search concludes. Assume that predecessors are also generated and added to the `FRINGE` in alphabetical order.

2. **Basic Search Methods [19 points]**

[*Warning: Facts and numbers used in the following problem are historically incorrect. Some are pure fiction. Stick to the problem as formulated below, not to history*]

Sergio Lopez was a rich ship owner living in Spain in the 17th century. In 1650, one of his ships was sunk in the Caribbean Sea with a huge cargo of gold. In 2008, an international company, Gold-Retriever, found the wreckage of the ship and brought back the gold worth $ 50 million. A month later, a Spanish citizen, named Alberto Lopez claimed that he was a direct descendent of Sergio and that, therefore, he was the owner of the gold. To prove his point he hired an attorney and asked him to collect all birth certificates showing that he is a direct descendant of Sergio.

We will assume that Spain has kept archives of all birth certificates during the last four centuries and that all Alberto's ancestors up to Sergio lived in Spain. We will also assume that the population of Spain between Sergio's birth and the present time has monotonically increased from about 6 million people to about 41 million today.

Let `Parents(`x`)` be the function that returns the two parents of an individual x (and their dates of birth) and `Children(`y`)` the function that returns the children of an individual y (and their dates of birth). We will finally assume that a birth certificate can be retrieved as easily by either knowing the name of a child or the name of one of his/her parents. In other words, it will cost the attorney the same amount of time to evaluate either function `Parents( )` or function `Children( )`.

(a) [**7 points**] Formulate the Attorney's task to prove that Alberto is a descendant of Sergio as a search problem:

  i. [**2 points**] Give a state representation for this problem.

  ii. [**1 points**] What is the initial state in your formulation?

  iii. [**3 points**] What are the available actions from a state and how does each modify the state in your formulation?

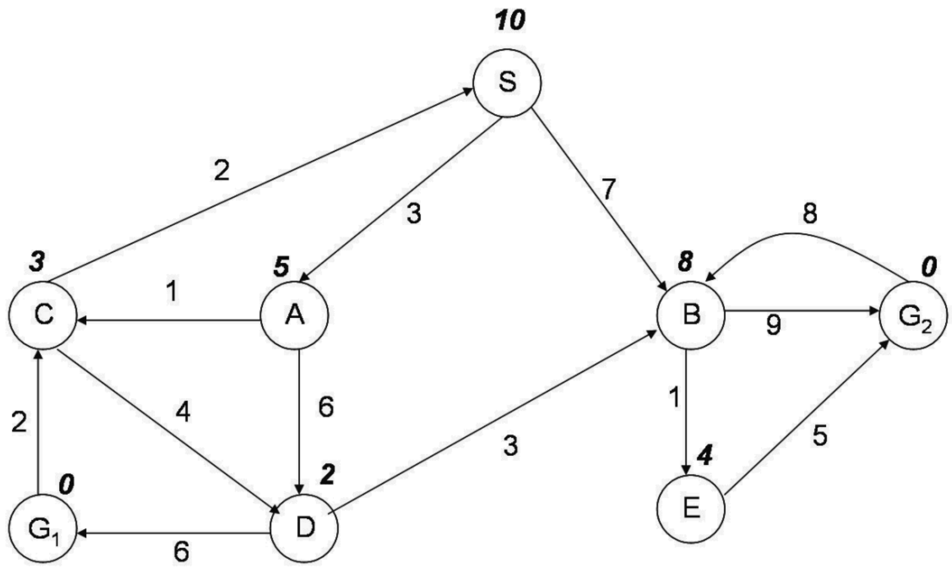
  iv. [**1 points**] What is the goal state in your formulation?

(b) [**4 points**] Is your formulation the only one possible for this problem? If your answer is 'yes', the explain why there is only one. If your answer is 'no', then give another one.

(c) [**4 points**] What would be the easier way for the attorney to proceed: with an initial state of Alberto, or an initial state of Sergio? Why?

(d) [**4 points**] Which search technique: breadth-first, depth-first, bi-directional, depth-limited, iterative deepening, etc. would you recommend to the attorney? Justify your answer. [There might not be a clear-cut answer.]

3. **Best-first Search [27 points]** Consider the following state graph:



The state space consists of states S, A, B, C, D, E, G_1, and G_2. S is the initial state and G_1 and G_2 are the goal states. The possible actions between states are indicated by arrows. So, the successor function for state S returns $\{A, B\}$; for A it returns $\{C, D\}$, etc .... The number labeling each arc (roughly at the mid-point of the arc) is the actual cost of the action. For example, the cost of going from S to A is 3. The number in bold italic near each state is the value of the heuristic function h at that state. For example, the value of h at state C is 3.

(a) **[24 points]** Run the following best-first search algorithms on the described search problem. For each, report the order that nodes are expanded by each algorithm, as well as the final path returned by each algorithm. For each algorithm **do not** check if a state is a re-visited one or not (hence, there may be several nodes pointing to the same state in the search tree). Each algorithm should terminate only when it removes a goal node from the fringe. If there is a tie in the priority, break it alphabetically by state, and then by which node was added to the fringe first (assume nodes are generated by the successor function in alphabetical order).

  i. **[8 points]** Uniform-cost search

  ii. **[8 points]** Greedy best-first search

  iii. **[8 points]** A* search

(b) **[3 points]** Is the heuristic function h defined by values provided in the figure admissible? How do you know? How does this affect each of the searches?

4. **Approximately optimal search [40 points]**

   The two objectives of finding a solution as quickly as possible and finding an optimal solution are often conflicting. In some problems, one may design two heuristic functions h_A and h_N, such that h_A is admissible and h_N is not admissible, with h_N resulting in much faster search most of the time. Then, one may try to take advantage of both functions.

   (a) **[30 points]** A best-first search algorithm called A^*_ϵ uses the evaluation function $f(N) = g(N) + h_A(N)$. At each iteration, A^*_ϵ expands a node N' such that

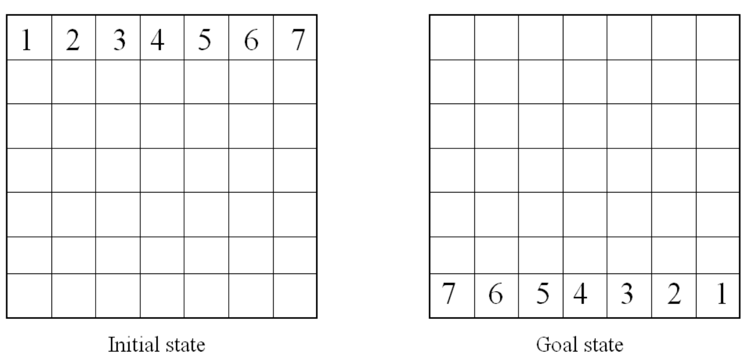
   $$f(N') \leq (1+\epsilon)\left[\min_{M \in \text{Fringe}} \{f(M)\}\right]$$

   where ϵ is some strictly positive number. (In other words, instead of guaranteeing that it will expand the node with minimum f-value in the fringe, it only guarantees that the node it expands will have an f-value less than $(1+\epsilon)$ times the minimum f-value found in the fringe.)

   What can you guarantee about the cost of the solution returned by A^*_ϵ? Your answer should be a proof, following the form of the proof of A^*'s optimality that we covered in class.

   (b) **[10 points]** Explain briefly how A^*_ϵ could use the second heuristic function h_N to reduce the time of the search. What tradeoff is being made in choosing ϵ? (Hint: think about what happens when there are many nodes that A^*_ϵ could return. Additionally, consider what happens to A^*_ϵ as ϵ gets closer to zero or when ϵ is very large.)

5. **EXTRA CREDIT: Vehicles on a grid [20 points]** A group of n vehicles, identified by $1, 2, \ldots, n$, can move in an $n \times n$ grid of squares. Initially, each vehicle i is located in square $(1, i)$ [meaning on 1st row and i-th column]. Its goal is to move to square $(n, n-i+1)$ [hence, the vehicles must move to a reverse order in the n-th row of the grid].



Example grid with n = 7

At each step, each of the n vehicles can move by one square up, down, left or right, or stay in the same square. If a vehicle does not move, one adjacent vehicle (but no more than one) can hop over it. No two vehicles are allowed to be in the same square at the end of any given step.

(a) **[5 points]** Formulate this problem as a search problem.

  i. **[3 points]** Give a specific, detailed representation or format for a generic state.

  ii. **[1 points]** Write down exactly what the initial state is, according to your state representation.

  iii. **[1 points]** Write down exactly what the goal state is, according to your chosen state representation.

(b) **[3 points]** What is the approximate size of the state space for this problem? Briefly explain your choice.

  i. n^2

  ii. n^3

  iii. n^{2n}

  iv. n^{2^n}

  v. n^{n^2}

(c) **[3 points]** What is the approximate branching factor of the search tree? Briefly explain your choice.

  i. 5

  ii. $5n$

  iii. 5^n

  iv. n^5

(d) **[5 points]** Suppose that vehicle i is located at square (r_i, c_i). What is the minimum number h_i of steps that this vehicle must perform to reach its goal square assuming that there are no other vehicles on the grid? Express h_i in mathematical form as a function of r_i, c_i, i, and n.

(e) **[4 points]**

  i. **[2 points]** Is $\min\{h_1, \ldots, h_n\}$ an admissible heuristic for the problem of moving all n vehicles to the goal squares (when the cost of a solution is the number of steps to reach the goal)? Justify your answer.

  ii. **[2 points]** Is $h_1 + \ldots + h_n$ an admissible heuristic? Justify your answer.