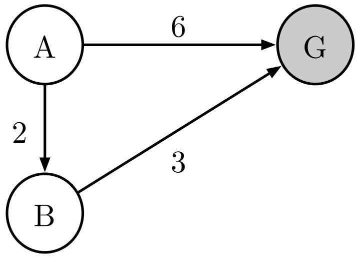
Q1. Search: Heuristic Function Properties

For the following questions, consider the search problem shown on the left. It has only three states, and three directed edges. A is the start node and G is the goal node. To the right, four different heuristic functions are defined, numbered I through IV.



	h(A)	h(B)	h(G)
I	4	1	0
II	5	4	0
III	4	3	0
IV	5	2	0

(a) Admissibility and Consistency

For each heuristic function, circle whether it is admissible and whether it is consistent with respect to the search problem given above.

	Admissible?		Consistent?	
I	Yes	No	Yes	No
II	Yes	No	Yes	No
III	Yes	No	Yes	No
IV	Yes	No	Yes	No

(b) Function Domination

Recall that *domination* has a specific meaning when talking about heuristic functions.

Circle all true statements among the following.

1. Heuristic function III dominates IV.
2. Heuristic function IV dominates III.
3. Heuristic functions III and IV have no dominance relationship.

4. Heuristic function I dominates IV.
5. Heuristic function IV dominates I.
6. Heuristic functions I and IV have no dominance relationship.

Q2. Search

Each True/False question is worth 1 points. Leaving a question blank is worth 0 points. **Answering incorrectly is worth −1 points.**

(a) Consider a graph search problem where for every action, the cost is at least ϵ, with $\epsilon > 0$. Assume the used heuristic is consistent.

(i) [*true* or *false*] Depth-first graph search is guaranteed to return an optimal solution.

(ii) [*true* or *false*] Breadth-first graph search is guaranteed to return an optimal solution.

(iii) [*true* or *false*] Uniform-cost graph search is guaranteed to return an optimal solution.

(iv) [*true* or *false*] Greedy graph search is guaranteed to return an optimal solution.

(v) [*true* or *false*] A* graph search is guaranteed to return an optimal solution.

(vi) [*true* or *false*] A* graph search is guaranteed to expand no more nodes than depth-first graph search.

(vii) [*true* or *false*] A* graph search is guaranteed to expand no more nodes than uniform-cost graph search.

(b) Let $h_1(s)$ be an admissible A* heuristic. Let $h_2(s) = 2h_1(s)$. Then:

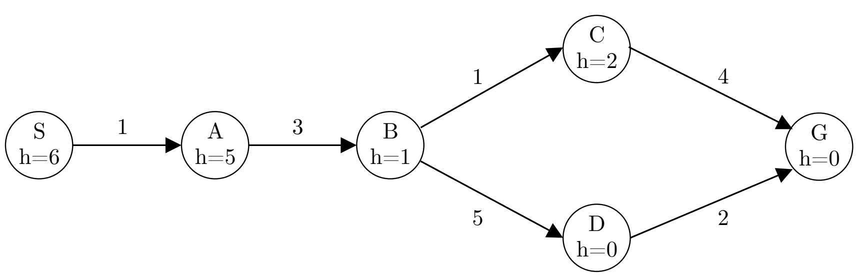
(i) [*true* or *false*] The solution found by A* tree search with h_2 is guaranteed to be an optimal solution.

(ii) [*true* or *false*] The solution found by A* tree search with h_2 is guaranteed to have a cost at most twice as much as the optimal path.

(iii) [*true* or *false*] The solution found by A* graph search with h_2 is guaranteed to be an optimal solution.

(c) The heuristic values for the graph below are not correct. For which single state (S, A, B, C, D, or G) could you change the heuristic value to make everything admissible and consistent? What range of values are possible to make this correction?

State: ☐ Range: ☐



Q3. Formulation: Holiday Shopping

You are programming a holiday shopping robot that will drive from store to store in order to buy all the gifts on your shopping list. You have a set of N gifts $G = \{g_1, g_2, \dots g_N\}$ that must be purchased. There are M stores, $S = \{s_1, s_2, \dots s_M\}$ each of which stocks a known inventory of items: we write $g_k \in s_i$ if store s_i stocks gift g_k. Shops may cover more than one gift on your list and will never be out of the items they stock. Your home is the store s_1, which stocks no items.

The actions you will consider are travel-and-buy actions in which the robot travels from its current location s_i to another store s_j in the fastest possible way and buys whatever items remaining on the shopping list that are sold at s_j. The time to travel-and-buy from s_i to s_j is $t(s_i, s_j)$. You may assume all travel-and-buy actions represent shortest paths, so there is no faster way to get between s_i and s_j via some other store. The robot begins at your home with no gifts purchased. You want it to buy all the items in as short a time as possible and return home.

For this planning problem, you use a state space where each state is a pair (s, u) where s is the current location and u is the set of unpurchased gifts on your list (so $g \in u$ indicates that gift g has not yet been purchased).

(a) How large is the state space in terms of the quantities defined above?

(b) For each of the following heuristics, which apply to states (s, u), circle whether it is admissible, consistent, neither, or both. Assume that the minimum of an empty set is zero.

(neither / admissible / consistent / both)	The shortest time from the current location to any other store: $\min_{s' \neq s} t(s, s')$
(neither / admissible / consistent / both)	The time to get home from the current location: $t(s, s_1)$
(neither / admissible / consistent / both)	The shortest time to get to any store selling any unpurchased gift: $\min_{g \in u}(\min_{s' : g \in s'} t(s, s'))$
(neither / admissible / consistent / both)	The shortest time to get home from any store selling any unpurchased gift: $\min_{g \in u}(\min_{s' : g \in s'} t(s', s_1))$
(neither / admissible / consistent / both)	The total time to get each unpurchased gift individually: $\sum_{g \in u}(\min_{s' : g \in s'} t(s, s'))$
(neither / admissible / consistent / both)	The number of unpurchased gifts times the shortest store-to-store time: $\lvert u \rvert(\min_{s_i, s_j \neq s_i} t(s_i, s_j))$

You have waited until very late to do your shopping, so you decide to send an swarm of R robot minions to shop in parallel. Each robot moves at the same speed, so the same store-to-store times apply. The problem is now to have all robots start at home, end at home, and for each item to have been bought by at least one robot (you don't have to worry about whether duplicates get bought). Hint: consider that robots may not all arrive at stores in sync.

(c) Give a minimal state space for this search problem (be formal and precise!)