ECE 479/579 **Name:**
Spring 2024 **Please type in upper case letters.**

February 27, 2024
Midterm Examination

1. (10 points) True or False

   a. Breadth-first search always expands at least as many nodes as A* search with a heuristic function h(n) that is an underestimate of the actual distance/cost from node n to G (goal) (h(n) <= h*(n)). **False**
   b. The major drawback of hill climbing is that it may not find the optimal solution True.
   c. An advantage of a greedy search is that it requires little memory when you implement it. **True**
   d. A depth-first search will always find an optimal solution if one exists and assuming that the state space is finite. **False**

2. (12 points) Consider the following problem statement: we would like to control a thermostat for a furnace. The thermostat's states include a temperature setting (TS), a temperatire reading (TR), a value of a state of the furnace (SSF) (either ON or OFF).

   The thermostat has three actions: it can turn the furnace on, turn it off, or do nothing. It can turn the furnace on if the ambient temperature is < 1 degree than the setting. It turns the furnace off whenever the ambient temperature is > 1 degree higher than the setting. Otherwise, it takes no action.

   You are to: **a) show what the state space is (data base DB), b) what the operators/rules are (Rules), and c) how you would control the furnace using the DB and Rules.**

   Please ignore situations when the furnace or thermostat might be faulty (for example, if the thermostat might be calling for heat but the furnace is not heating etc.).

   As the initial state assume any valid values for example 70, 70.5, and OFF. But you need not to work with any specific values to define the AI production system, specifically the DB, Rules and Control Strategy.

DB = set of states, where the state is defined as s = (TS, TR, SSF)

Rules:

R1: if (SSF = OFF or SSF = ON) and TS = x and TR = y and (x – 1 < y < x + 1)
Then DO NOTHING

R2: if (SSF = OFF) and TS = x and TR = y and y < x – 1
then SSF = ON

R2: if (SSF = ON) and TS = x and TR = y and y > x + 1
then SSF = OFF

Control strategy: (set up a forever loop)

LOOP Forever

R1: if (SSF = OFF or SSF = ON) and TS = x and TR = y and (x – 1 < y < x + 1
Then DO NOTHING (do not switch state)

R2: if (SSF = OFF) and TS = x and TR = y and y < x – 1
then SSF = ON

R2: if (SSF = ON) and TS = x and TR = y and y > x + 1
then SSF = OFF

END LOOP

3. (5 points) In A*, what is the reason to use a heuristic function that underestimates the real cost from a node open for state expansion to the goal node?

Answer: the underestimate guarantees that once a path has been found from the initial state to the goal node with the cost that is <= to $f(n) = g(n) + h(n)$ for all n that are still open for state expansion, then the search can terminate, and the path found is the best one. (We assume costs are non-negative.)

4. (12 points) In class, we have discussed the A* algorithm and its application to the 8-puzzle problem. The merit of a node was defined as f(n) = g(n) + h(n) where h(n) = number of misplaced tiles with respect to the goal state description (ignore the blank) and g(n) = depth of the node n in the search tree.

There are two parts to this question:

a. Suppose we were to use h(n) = 0 as the estimate of the remaining cost of the path (number of moves) from n to G. What sort of search would A* become?

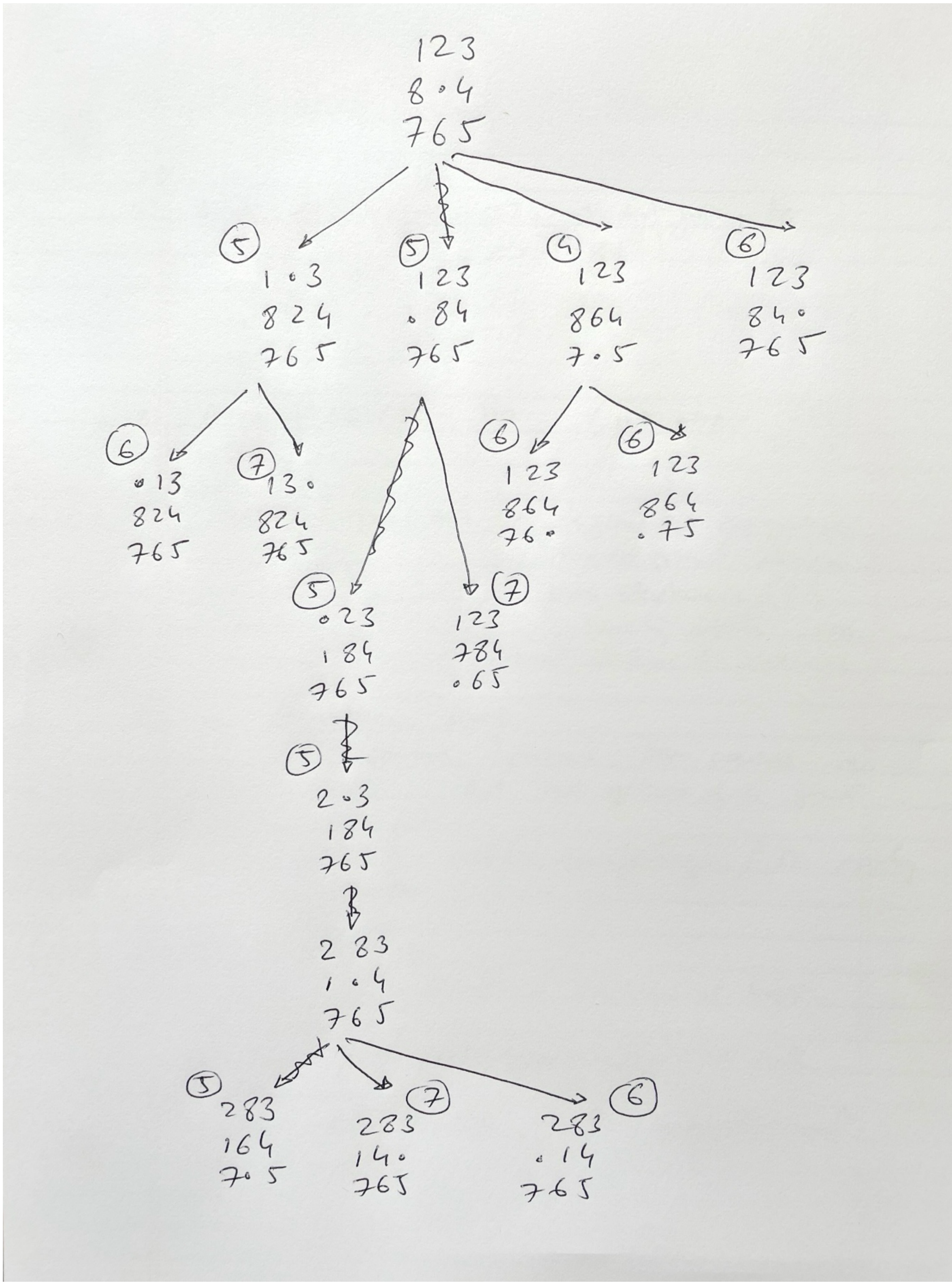
ANSWER here: **Branch-and-Bound with dynamic programming.**

b. Apply A* to the following set of states. **SHOW** the generated search tree and **the solution path**.

2 8 3	1 2 3
1 6 4	8 * 4
7 * 5 (**GOAL STATE**) !!!	7 6 5 (**INITIAL STATE**) !!!

USE NEXT PAGE FOR SOLUTION OF PART

This page for solution of problem 4b



5. (10 points): **Graduate students only (UG for extra credit) Two parts:**

a) Define the monotone restriction (MR) on the h(n) function in A* search.
b) Does the following function: h(n): number of misplaced tiles (with respect to the goal state description) in the 8-puzzle problem, satisfy the MR property?

Please justify your "yes" or "no" answer.

Answer:

a. For any two ni and nj such that nj is a successor of ni,
h(ni) <= c(ni, nj) + h(nj)
where c(ni, nj) is the actual cost from ni to nj.

b. Assume h(ni) = k at node ni, h(nj) = m at nj

m is the no. of misplaced tiles at nj. A move's cost is 1.

Thus, when a move that decreases the number of misplaced tiles is executed m = k – 1. Therefore

k <= 1 + (k – 1) holds true.

If you make a move that does not change the no. of misplaced titles

k <= 1 + m holds true as k = m

If a move is made that worsens the situation

m = k + 1, thus
k <= 1 + (k +1) holds true