Write **clearly** and **in the box**:

CSCI 3202
Midterm Exam
Remote edition
Fall 2020

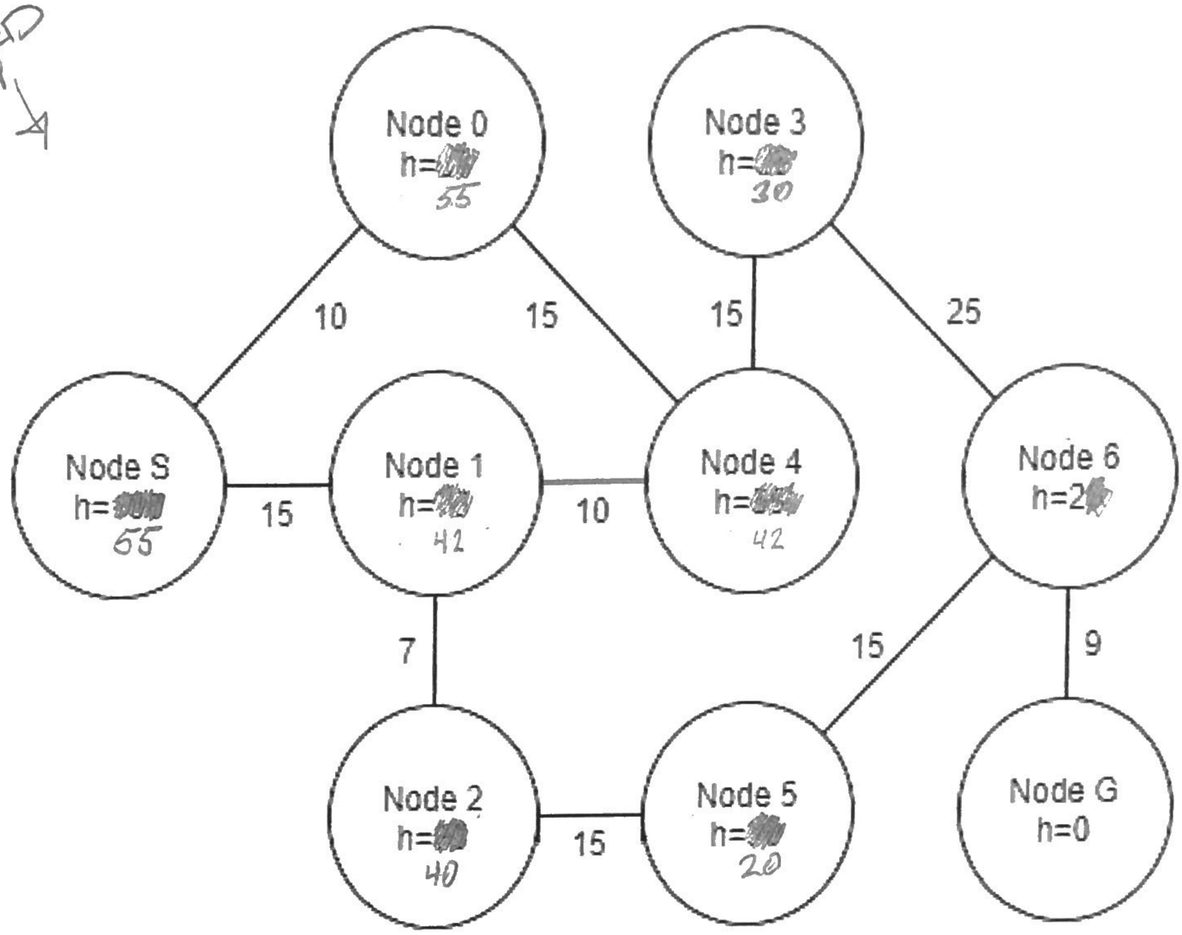
| | |
|---|---|
| **Name:** | Jacob (Jake) Henson |
| **Student ID:** | 105963531 |
| **Section number:** | 001 |

- **RIGHT NOW!** Include your name, student ID and section number on the top of your exam. If you're handwriting your exam, write this information at the top of the first page!
- You may use the textbook, your notes, lecture materials, and Piazza as recourses. Piazza posts should not be about exact exam questions, but you may ask for technical clarifications and ask for help on review/past exam questions that might help you. You may not use external sources *from the internet or collaborate with your peers.*
- You may use a calculator.
- If you print a copy of the exam, clearly mark answers to multiple choice questions in the provided answer box. If you type or hand-write your exam answers, write each problem on their own line, clearly indicating both the problem number and answer letter. Start each new problem on a new page.
- Mark only one answer for multiple choice questions. If you think two answers are correct, mark the answer that **best** answers the question. No justification is required for multiple choice questions. For handwriting multiple choice answers, clearly mark both the number of the problem and your answer for each and every problem.
- For free response questions you must clearly justify *all* conclusions to receive full credit. A correct answer with no supporting work will receive no credit.
- The Exam is due to Gradescope by midnight on Monday, October 12.
- When submitting your exam to Gradescope, use their submission tool to mark on which pages you answered specific questions.

| Problem | Student Points | Max Points |
|---|---|---|
| Discrete Search | | 35 |
| Games | | 30 |
| Short Response | | 20 |
| Information | | 15 |
| Total | | 100 |

(1) **[35 points]** For problem 1, refer to the following graph, with heuristic values $h(n)$ depicted inside each node in parenthesis. Consider the task of finding the shortest path from **S** to **G**, where step costs to travel between two nodes are given as edge weights.

UPDATED GRAPH




1A) **[5 points]** In what order would BFS **expand** the nodes of the graph? List all nodes that are expanded and the order in which they are expanded. Assume that numeric nodes are expanded lowest-first when an option exists.

0. E:{} F:{S}
1. E: {S} Fr: {0,1}
2. E: {S,0} Fr: {1,4}
3. E: {S,0,1} Fr: {4,2}
4. E: {S,0,1,2} F: {4,3}
5. E: {S,0,1,2,4} F: {3,5}
6. E: {S,0,1,2,4,3} F: {6,5}
7. E: {S,0,1,2,4,3,5} F: {6}

END: {S,0,1,2,4,3,5,6,G}

1B) **[5 points]** In what order would DFS **expand** the nodes of the graph? List all nodes that are expanded in the order in which they are expanded. Assume that numeric nodes are expanded lowest-first when an option exists.

1. {S}
2. {S,0}
3. {S,0,4}
4. {S,0,4,1}
5. {S,0,4,1,2}
6. {S,0,4,1,2,5}
7. {S,0,4,1,2,5,6}
8. {S,0,4,1,2,5,6,Goal}

1C) **[5 points]** In what order would UCS **expand** the nodes of the graph? List all nodes that are expanded in the order in which they are expanded.

1. {(S,0)}
2. {(S,0),(0,10)}
3. {(S,0),(0,10),(1,15)}
4. {(S,0),(0,10),(1,15),(4,25)}
5. {(S,0),(0,10),(1,15),(4,25),(2,22)}
5. {(S,0),(0,10),(1,15),(4,25),(2,22),(5,37)} ...

cont. on next page

1D) **[5 points]** In what order would A* **expand** the nodes of the graph? List all nodes that are expanded in the order in which they are expanded.

...

1C cont.) 6. {(S,0),(0,10),(1,15),(4,25),(2,22),(5,37)}

NON-OPTIMAL WAY OF REACHING 6; EXCLUDE

7. {(S,0),(0,10),(1,15),(4,25),(2,22),(5,37),(3,50)}

8. {(S,0),(0,10),(1,15),(4,25),(2,22),(5,37),(3,50),~~(6,75)~~,(6,52)}

9. {(S,0),(0,10),(1,15),(4,25),(2,22),(5,37),(3,50),(6,52),(G,61)}

TOTAL = {S, 0, 1, 4, 2, 5, 3, 6, G}

1D.) A* $f(n) = g(n) + h(n)$

1. {(S, g=0, h=55, f=55)}
2. {(S, ...), (1, g=15, h=42, f=57)}
3. {(S, ...), (1, ...), (0, g=10, h=55, f=65)}
4. {(S, ...), (1, ...), (0, ...), (2, g=22, h=40, f=62)}
5. {(S, ...), (1, ...), (0, ...), (2, ...), (5, g=37, h=20, f=57)}
6. {(S, ...), (1, ...), (0, ...), (2, ...), (5, ...), (6, g=52, h=2, f=54)}
7. {(S, ..), (1, ...), (0, ..), (2, ...), (5, ...), (6, ...), (G, g=63, h=0, f=63)}

TOTAL: {S, 1, 0, 2, 5, 6, G}

1E) **[5 points]** Is the heuristic $h$ on the graph admissible? If so, explain why. If not, explain for which single state $n$ you could change the values of $h(n)$ to make everything admissible and consistent. What range of values for $h$ are possible to make this correction?

The heuristic on the UPDATED graph is NOT admissable because Node 2 has a heuristic of 4Ø, which overestimates the actual cost of reaching the goal (39). For h(2) (Node 2) values between 39 - 21 would make heuristic h admissable!

1F) **[4 points]** Suppose after making any adjustments that you've arrived at a consistent and admissible heuristic $h_1(n)$. You then decide to use the new heuristic $h_2(n) = 2h_1(n)$. Is $A^*$ the solution found using $h_2$ *guaranteed* to be optimal? Justify your answer.

Because $h_2(n)$ could be an overestimate, it is therefore non-admissable and as such will no longer necessarily produce an optimal solution (i.e. not garunteed)

1G) **[3 points]** Provide a reason as to why using an inadmissible or inconsistent heuristic may be a bad idea. A* may not provide an optimal path if given an inconsistent heuristic, and as such, may give you a path which is NOT the best one. (i.e, a heuristic which doesn't properly assign weight could pick the wrong path)

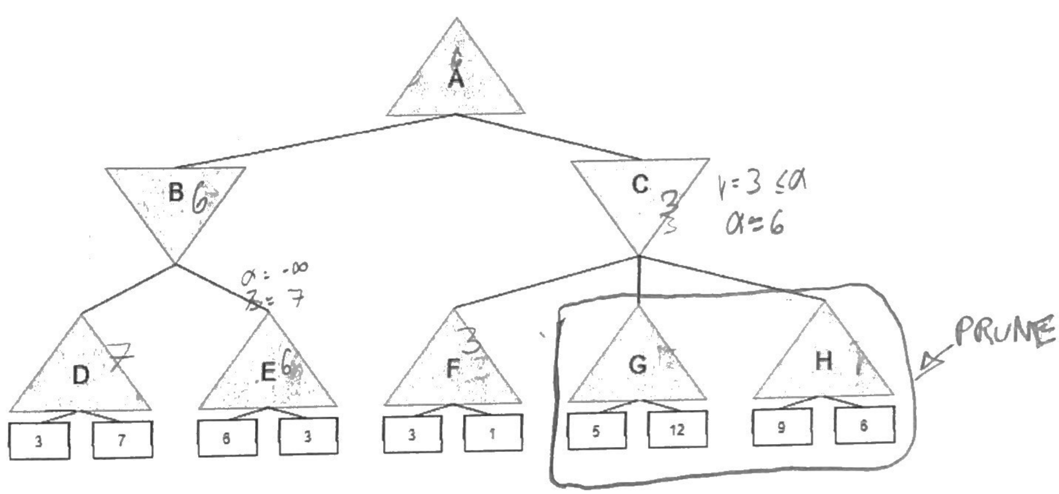
1H) **[3 points]** Provide a reason as to why using an inadmissible or inconsistent heuristic may be a good idea.

For something like relaxed heuristics, it could be the case that a non-admissible heuristic may produce a result which dominates another so it may be a good idea to use non-admissible because it could provide a more optimal solution

2C) **[4 points]** On the shallow game tree of depth 2 from part 2B, what would we infer is the value of the game? Use the score of "number of warp drives reached by the player" as your evaluation function. I think the value would simply be 0 here... this is because there is no outcome on the depth of 2 tree that produces a "win" condition (warp drives reached by player = 0).

2D) **[4 points]** If we were to consider a game tree of depth 16 - holding eight moves for each player instead of only one - what would minimax compute as the value of the starting position of the game? Explain your reasoning. I think minimax would produce a value of 1 because either the player has won (1) or "lost" (0) (the game doesn't result in a win). Because the starting position by default chooses max between 1 and 0, this results in 1

2E) **[8 points]** Consider the minimax tree below. Upward pointing triangles are MAX nodes, downward pointing triangles are MIN nodes.




What are the minimax values associated with each node? Do not consider any pruning at this point.

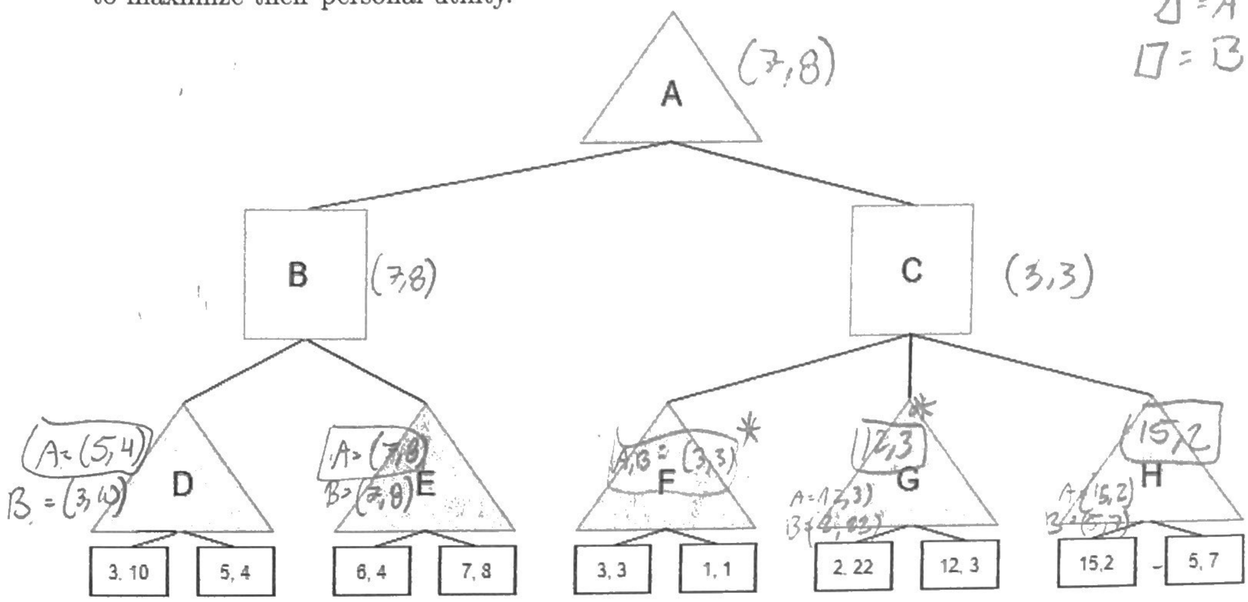
**A:** 6 **B:** 6 **C:** 3 **D:** 7

**E:** 6 **F:** 3 **G:** 12 **H:** 9

On the diagram of the tree, indicate clearly which branches/leaves are pruned when alpha-beta pruning is applied to this game tree. Assume that nodes are expanded from left-to-right at each layer. Briefly justify how you know those branches/leaves can be pruned. Vague responses along the lines of simply saying "alpha-beta pruning algorithm" will receive 0 points. The reason why the algorithm would prune G and H is because at C, the value is 3, and this is less than α (which is 6) so the max would never let the game go to G or H because it's going to pick 6 anyway!

2F) **[5 points (EXTRA CREDIT)]**. Consider a non zero-sum representation of a game. In this setup, player A's utility is denoted as the first of two leaf numbers and player B's utility by the second. Fill in this non-zero game tree assuming that each player is acting optimally to maximize their personal utility.

Δ = A
□ = B

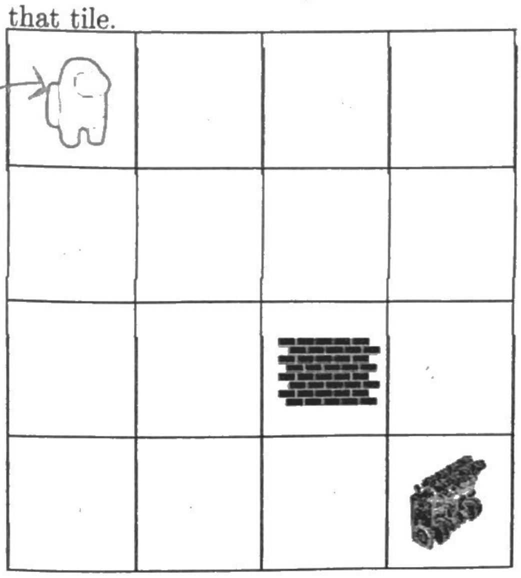



A = 7,8
because A will pick the highest of the two between (7,8) and (3,3)

* Between F and G, I think Player B would pick F arbitrarily, because 3 ≥ 3 so it has no reason to pick otherwise.

(2) **[30 points]** Parts A-D refer to this game. We're playing our favorite new game, "walls among us." In this game, we play as a noble space explorer, maintaining our 2-dimensional ship represented by the 4x4 grid below. On our turn, we may move in one of the 4 cardinal directions, and must move into an unoccupied square. Our opponent is attempting to sabotage us, and has sabotaged the ship's warp drive! We have to fix it, but our opponent is in the game, and plays as a brick wall trying to block our way. On the wall's turn, they also must move in one of the 4 cardinal directions, and must also move to an unoccupied square. The game ends - with us as the victor - when we move to the goal in the bottom-right corner of the map. The wall may never move to that tile.

IDK he seems kinda sus...




2A) **[4 points]** Assuming the engine might be moved to any tile on the board, what is the size of the overall state space? The exact result should use the fact that the wall can never overlap with either you or the engine.

N/E/S/W × 4 dir

4x4 spaces = 16, can be empty, engine, wall, astro = $4^{16}$

Astro = Any of 15 tiles · 4 directions = $15 \cdot 4$ (no overlap w/ wall)

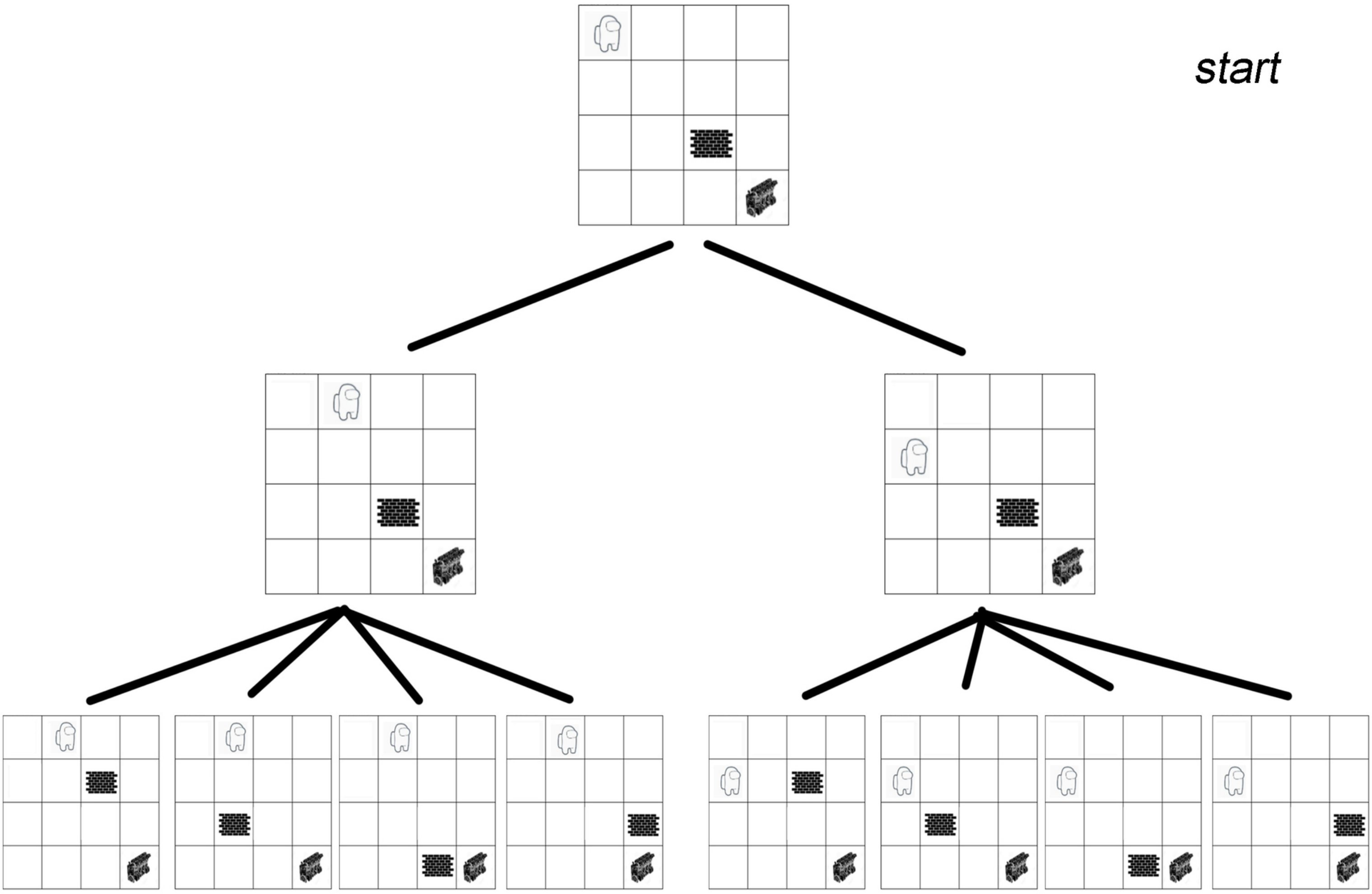
Engine = Any of 15 tiles = 15

Wall = Any of 14 tiles · 4 directions = $14 \cdot 4$

$\boxed{\text{Total: } = 4^{16} \cdot 15^2 \cdot 4 \cdot 14 \cdot 4}$

2B) **[10 points]** Draw a game tree with one move for each player from the starting position shown. Nodes in the tree represent game states (location of all agents and walls). Edges in the tree connect successor states to their parent states. Draw only the legal moves.

DOING THIS DIGITALLY CAUSE ITS GONNA TAKE TOO LONG TO DO MANUALLY BY HAND



start

*note - this game tree assumes the Astronaut is going first

(3) **[20 points]** Other Searches. For each question, answer the prompt and provide at least one full sentence of reasoning and justification. More than one sentence may be necessary.

3A) **[4 points]** In what situation(s) is Breadth First Search identical to Uniform Cost Search?

In situations where there are an extremely small amount of nodes with little to no choices, (i.e. straight line) but also in situations of equal cost between all choices. this is because UCS+BFS only evaluate based on what's directly in front of them at each step.

3B) **[4 points]** Is hill climbing guaranteed to find the optimal solution? Why or why not?

Not necessarily - it is not garunteed to find the best possible optimal solution because it is limited to finding local maxima based on the neighbors, which may or may not be the optimal solution for the whole problem! (global maxima)

3C) **[12 points]** Suppose we were interested in using the genetic algorithm to find a solution or solutions to the system given by: $a + 5b + 6c - 12d + e = 1204$ for positive integers $a, b, c, d, e$. Suppose your initial population consists of 100 lists of the form $[a, b, c, d, e]$, where each term is initialized as a random integer from 1 to 500.

Propose a working genetic algorithm for this problem. This includes: a fitness function to minimize - *so one that will return zero when at a valid solution* - and a selection/breeding/mutation scheme that will find valid solutions to the problem.

**Proposed Fitness Function:** Determine the fitness score, should add together the fitness of each to get a total fitness score for each list: (i.e., Fitness[a]+ Fitness[b]+...). From here, caclulate each member's probability of reproduction based on an objective function's performance. In this example, each element's Performance[X] / Total Fitness for each list. So you'd have a probability score for each!

**Proposed $P(breeding)$ Function:**

Evaluate the probability of reproducing/breeding based on fitness score, select ones with high fitness scores and assign a probability of reproduction based on the full list of each, I.e., for the top 20, with highest fitness, randomly pick 2 parents.

**Proposed Breeding/Mutation Scheme:**

Select two parents, and use randomization to pick up traits from each parent. Create new list and repeat the process until done. In this example, you could for example select a random amount of 2-5 elements from each parent

4A) **[15 points]** Suppose we're going to catch the Buff Bus, and want to decide when to go wait for it. We decide to use the the linear loss function

$$L_l(d,x) = \begin{cases} 2(x-d) & x \geq d \\ 4(d-x) & x < d \end{cases}$$

.

4A) **[7 points]** We model the Buff Bus arrival times as an exponential random variable $X$ that arrives on average once per hour, so they have probability density function of $f(x) = e^{-x}$ for $x > 0$ (note: this has mean of $E_X[x] = 1$). Set up but don't evaluate an integral or integrals for the *expected* loss $E_X[L(d,x)]$.

pdf $= e^{-x}$ $\quad X > \emptyset$

$E[x] = \int_x x f(x)\,dx$

$$E_X[L(d,x)] = \int_{-\infty}^{\infty} L(d,x) f(x)\,dx \qquad \int_0^{\infty} x e^{-x}\,dx = 1$$

$$= \int_0^d 4(d-x)\,e^{-x}\,dx + \int_d^{\infty} 2(x-d)\,e^{-x}\,dx$$

I don't feel like we've been given enough info in class to solve this one

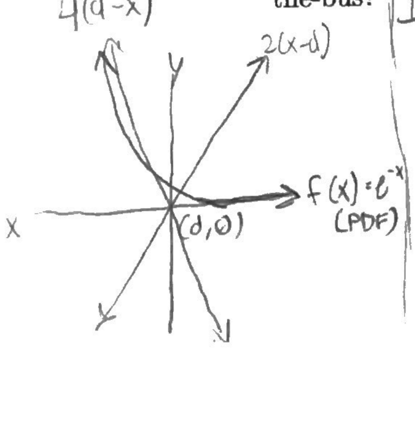
4B) **[4 points]** In class we discussed some conditions where the expected value of including uncertainty (EVIU) might be zero; do those apply here? Why or why not? What's larger in this problem, the expected value of perfect information (EVPI) or the EVIU? You do not need to calculate the exact values but you may if you wish.

$EVIU = E[L(d_{iu}, x) - L(d_{Bayes}, x)]$

EVIU would include random sample and would NOT be ∅ because EVIU is only ∅ when the Loss function is quadratic.

EVPI would likely just be 1 because it uses the mean of PDF, which in part A was determined to be 1. I think it would have a lower value because you'd hypothetically know when the bus shows up, meaning you'd have a lower expected value of loss.

4C) **[4 points]** In the loss function above, are we considering it more costly to overestimate or to underestimate? How do you know? Do you think this is a reasonable model for waiting-for-the-bus?




It would be more costly to underestimate because the loss function is steeper to the left of d and would be more costly because it would go underneath the PDF. (Probability of E[L] is larger)

This model is probably NOT reasonable because it factors in a linear loss function which doesn't allign with the PDF perfectly.