Name: ______________________________________ Andrew Id (Block Capitals): _____________________

ANSWERS TO 16-731/15-780 Midterm, Spring 2002

Tuesday Mar 12, 2002

1. Place your name and your andrew email address on the front page.
2. You may use any and all notes, as well as the class textbook. Keep in mind, however, that this midterm was designed in full awareness of such.
3. The maximum possible score on this exam is 100. You have 80 minutes.
4. Good luck!

1 Search Algorithm Comparison (15 points)

Let's define the `INFGRID` problem. In this problem, we have a robot in an infinitely large 2D grid world, and we wish to plan a path from the start location (x_s, y_s) to the goal location (x_g, y_g) that is a finite distance away. Possible moves are one step moves in any of the cardinal directions $\{North, South, East, West\}$, except that certain of the grid cells are obstacle cells that the robot cannot move into.

Assumptions:

- For each algorithm, assume that the successors function always generates successor states by applying moves in the same order $\{North, South, East, West\}$. We are not using backwards search, and there is no randomized component in any of the algorithms.
- Best-first search and A^* search both use the Manhattan distance heuristic. The heuristic value of a cell at position (x, y) is

$$h(x, y) = |x - x_g| + |y - y_g|$$

Questions:

(a) Is the heuristic h admissible? Just answer yes or no.

ANSWER: Yes (it is a lower bound on the actual number of moves a cell is from the goal).

(b) Fill in the table below with properties of some of our favorite search algorithms, when they are applied to `INFGRID`.

Instructions:

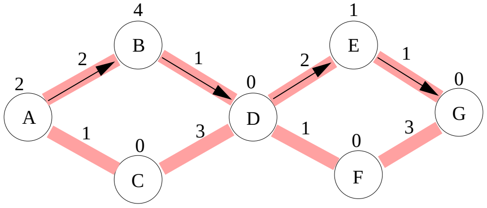
- The *Complete?* and *Optimal?* columns are yes or no questions. Mark them `Y` or `N` based on whether the algorithm has that property or not, when applied to `INFGRID`. Note: We say an incomplete algorithm is optimal iff it returns an optimal solution whenever it returns any solution (this is not necessarily a standard definition, but use it to fill out the *Optimal?* column for this question).
- For the *Memory usage* column, mark an algorithm `Low` if it uses memory $O(d)$, where d is the maximum depth of the search tree, and `High` if its memory usage is greater than $O(d)$. Of course, `Low` may still be infinite if d is not bounded, but don't worry about that.

ANSWER: Answers are in the table below.

Algorithm	*Complete?*	*Optimal?*	*Memory usage*
Breadth-first search	Y	Y	High
Depth-first search	N	N	Low
Depth-first iterative deepening	Y	Y	Low
Best-first search	Y	N	High
A^*	Y	Y	High

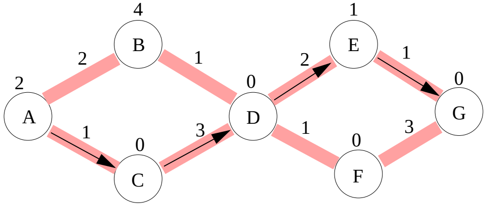
2 A^* Search (15 points)

The following is a graph that we are searching with A^*. Nodes are labeled with letters. Edges are the thick shaded lines. The number above each node is its heuristic value (e.g., $h(A) = 2$). The number above each edge is the transition cost (e.g., $cost(C, D) = 3$). You will see that the optimal path is marked for you with arrows.

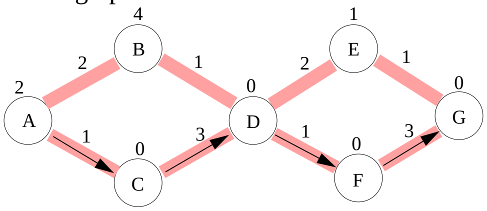


Questions: ANSWER: Answers are drawn in the graphs.

(a) Oops! Alice has implemented A^*, but her version has a mistake. It is identical to the correct A^*, except that when it visits a node n that has already been expanded, it immediately skips n instead of checking if it needs to reinsert n into the priority queue. Mark the path found by Alice's version of A^* in the graph below. Use arrows like the ones that show the optimal path above.

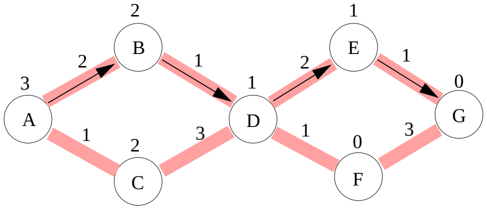


(b) Bob has also made a mistake. His version of A^* is identical to the correct A^*, except that it declares completion when it first visits the goal node G instead of waiting until G is popped off the priority queue. Mark the path found by Bob's version of A^* in the graph below:



(c) Carmen has implemented the same algorithm as Alice, but not by mistake. In addition to changing the algorithm, she changed the heuristic h so that it generates the values that you see in the graph below. With Carmen's new heuristic, Alice's algorithm is optimal, because the new heuristic has a special property we have discussed in class. What is the property?

ANSWER: The new heuristic is monotone (see assignment 1). Normal A^* only needs a heuristic to be admissible to find an optimal solution. Alice's modified version needs monotonicity, which is a stronger property (it implies admissibility). Both heuristics (the original one which caused the problem and the new one) are admissible.



3 Robot Motion Planning (10 points)

In the following configuration space, let

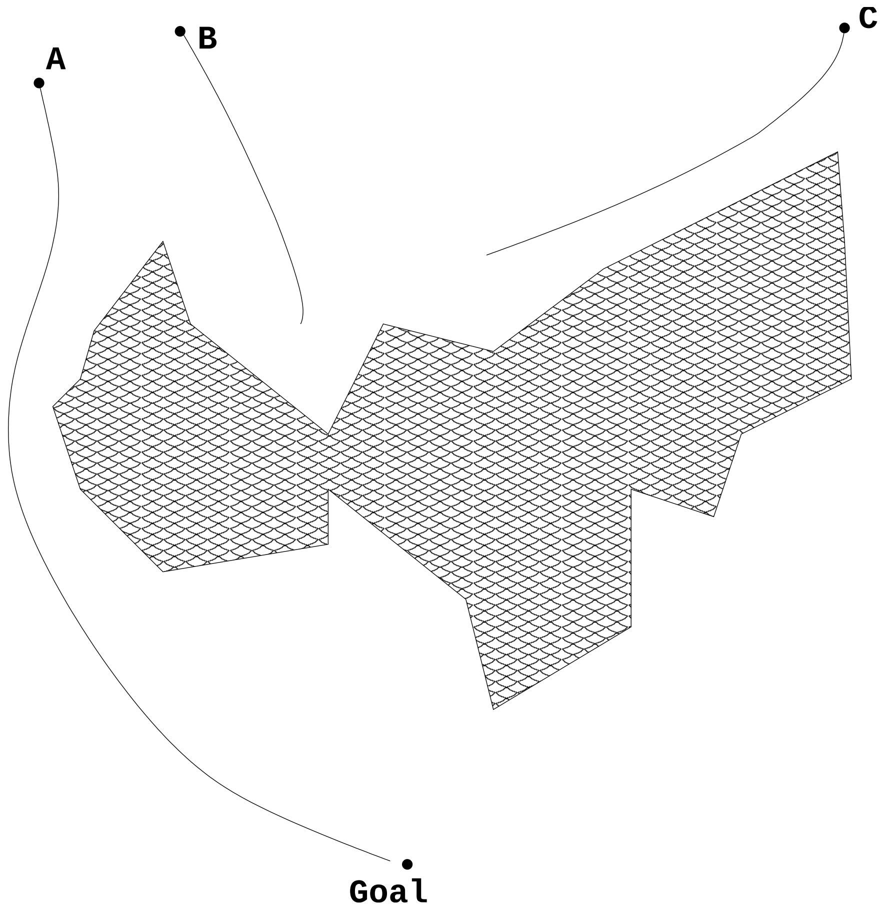
- d_0 = distance from robot to closest point on the obstacle in centimeters.
- d_g = distance from robot to the goal in centimeters.

Suppose the robot uses the potential field method of path planning, with the field value defined as $d_g + 1/d_o$.

(a) Draw (roughly) the path the the robot would take starting from point A on the diagram.

(b) Draw (roughly) the path the the robot would take starting from point B on the diagram.

(c) Draw (roughly) the path the the robot would take starting from point C on the diagram.



Note that B would begin by move towards goal, but the closest obstacle point would repel B slightly. Because that point is clearly to the left of the line between B and the goal, the repelling effect would shift the point to the right and thus push towards the local optimum. A similar argument applies to point C.

4 Constraint Satisfaction (10 points)

Here is a boolean satisfiability problem using the exclusive-or operator ($\otimes$). Note that in order for a set of variables to evaluate to 1 when they are exclusive-or'd together it is necessary and sufficient that an odd number of the variables have value 1 and the rest have value zero.

$$A \otimes B \otimes C$$
$$B \otimes D \otimes E$$
$$C \otimes D \otimes F$$
$$B \otimes D \otimes F$$

Suppose we run depth-first search in which the variables are ordered alphabetically (we try instatiating A first, then B etc). Suppose we try the value 0 first, then 1. Suppose that at the start we run constraint propagation, and suppose we also run full CP every time DFS instantiates a variable.

Which one of the following statements is true:

(i) The problem is solved (by CP) before we even need to start DFS

(ii) CP proves that the problem has no solution before we even need to start DFS

(iii) We do have to do DFS, but it solves the problem without ever needing to backtrack.

(iv) We do have to do DFS, but it proves the problem is insoluble without ever needing to backtrack.

(v) The first time we backtrack is when we try instantiating A to 0, and CP discovers an inconsistency

(vi) During the search we reach a point at which DFS tries instantiating B to 0, and then, when CP discovers an inconsistency, is the first time at which we backtrack.

(vii) During the search we reach a point at which DFS tries instantiating C to 0, and then, when CP discovers an inconsistency, is the first time at which we backtrack.

(viii) During the search we reach a point at which DFS tries instantiating D to 0, and then, when CP discovers an inconsistency, is the first time at which we backtrack.

(ix) During the search we reach a point at which DFS tries instantiating E to 0, and then, when CP discovers an inconsistency, is the first time at which we backtrack.

(x) During the search we reach a point at which DFS tries instantiating F to 0, and then, when CP discovers an inconsistency, is the first time at which we backtrack.

ANSWER: The correct answer is (viii). Instantiating A with 0 will not cause CP to do anything. Setting B=0 will cause CP to force C=1. Then when we try to make D=0, Constraint number 2 will force E=0, Constraint 3 will force F=0 and then constraint 4 will try to force F=1, causing a violation.

5 Simulated Annealing and Hill-climbing (10 Points)

Here is the pseudo-code for simulated annealing beginning in Configuration X and with initial temperature T and temperature decay rate r.

```
1. Let X :=initial object
2. Let E :=Eval(X)
3. Let X' :=randomly chosen configuration chosen from the moveset of X
4. Let E' :=Eval(X')
5. Let $z$:= a number drawn randomly uniformly between 0 and 1
6. If $E' > E$ or $\exp(-(E - E')/T) > z$ then
     • X :=X'
     • E :=E'
7. $T$:=$r \times T$
8. If a convergence test is satisfied then halt. Else go to Step 3.
```

(a) Normally r, the temperature decay rate, is chosen in the range $0 < r < 1$. How would the behavior of simulated annealing change if $r > 1$?

The change will always be accepted and we'll do a random walk.

(b) Alternatively, how would it change if $r = 0$?

Negative changes will never be accepted and we'll do randomized hill-climbing.

(c) If we simplified the conditional test in Step 6 to

If $\exp(-(E - E')/T) > z$ then

how would the behavior of simulated annealing change?

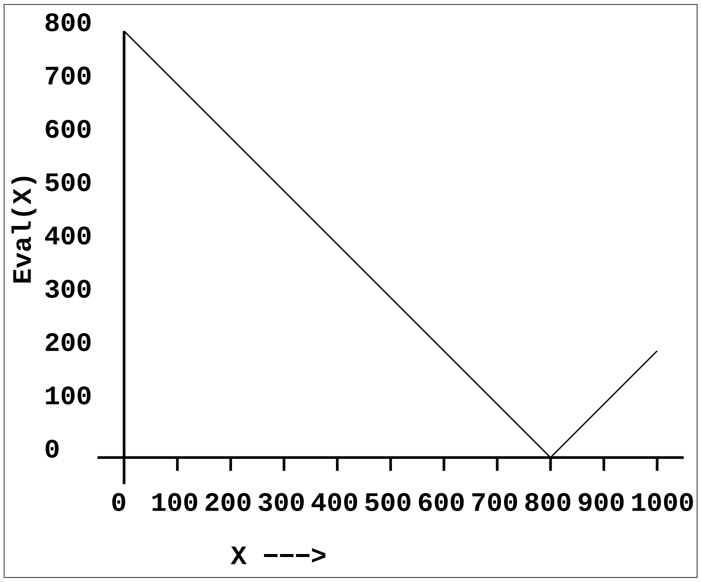
The algorithm would be unaffected, since if $E' > E$ the expression will certainly be greater than 1 and thus certainly greater than z.

Question Continues on next page

Suppose we are searching the space of integers between 1 and 1000. Suppose that the moveset is defined thus:

$$\begin{array}{lcc} \textsc{MoveSet}(X) = & \{1\} & \text{if } X = 0 \\ \textsc{MoveSet}(X) = & \{999\} & \text{if } X = 1000 \\ \textsc{MoveSet}(X) = & \{X-1, X+1\} & \text{otherwise} \end{array}$$

And suppose that *Eval*$(X) = |X - 800|$ so that the global optimum is at $X = 0$, when *Eval*$(X) = 800$. Note that there's a local optimum at $X = 1000$ when *Eval*$(X) = 200$. The function is graphed below:



(d) If we start hill-climbing search at $X = 900$ will it find the global optimum? (just answer yes or no)

No. It will hit the local optimum at X = 1000.

(e) If we start simulated annealing at $X = 900$ with initial temperature $T = 1$ and decay rate $r = 0.8$ is there better than a fifty fifty chance of reaching the global optimum within a million steps? (just answer yes or no)

No chance. After about 20 steps there will be essentially zero chance of moving left and so on each attempt we'll move right or almost certainly won't move. Thus we'll hit the X=1000 local optimum and will remain there.

6 Genetic Algorithms (10 points)

Suppose you are running GAs on bitstrings of length 16, in which we want to maximize symmetry: the extent to which the bitstring is a mirror image of itself (also known as being a palindrome). More formally:

Score = Number of bits that agree with their mirror image position.

Examples:

- Score(`1100110110110011`) = 16 (this is an example of an optimal bitstring)
- Score(`0000000011111111`) = 0
- Score(`0100000011111111`) = 2

Suppose you run GA with the following parameter settings:

- Single-point crossover
- Mutation rate = 0.01
- Population size 1000 (with an initial population of randomly generated strings)
- Stochastic Universal Sampling for selection (i.e. Roulette-wheel style)

Let N = the number of crossovers performed before an optimal bitstring is discovered.

Question: What is the most likely value of N? (note: we will accept any answer provided it is not less than half the correct value of N and provided it is not greater than twice the correct value of N).

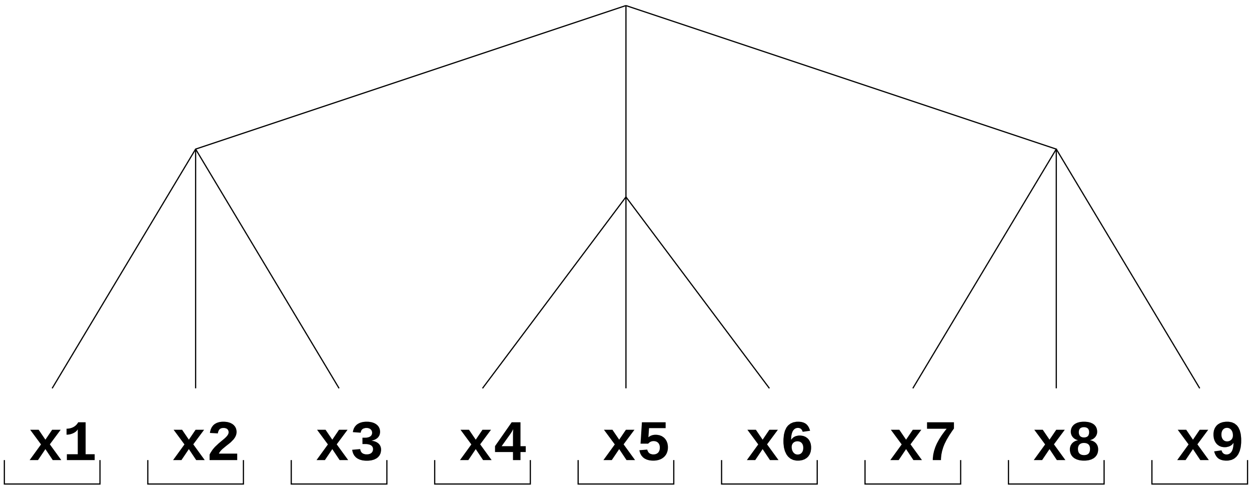
N = 0. One in 256 bitstrings will be optimal and so it is very likely that in the first 1000 strings generated at least one (and probably three or four) optimal strings will be generated. This is before any crossover has occured.

7 Alpha-beta Search (10 points)

The following diagram depicts a conventional game tree in which player A (the maximizer) makes the decision at the top level and player B (the minimizer) makes the decision at the second level.

We will run alpha-beta on the tree. It will always try expanding children left-to-right.

Your job is to fill in values for the nine leaves, chosen such that alpha-beta will not be able to do any pruning at all.



Any solution in which $\min(x_4, x_5) > \min(x_1, x_2, x_3)$ *and where* $\min(x_7, x_8) > \max(\min(x_1, x_2, x_3), \min(x_4, x_5, x_6))$.

A simple example is 1,1,1,2,2,2,3,3,3.

8 Optimal Auction Design (20 points)

Here is a nice general approach for running a one-item auction so that the auctioneer will make lots of money:

1. Ask each bidder i to secretly report its valuation v_i. This is how much the item is worth to the bidder.
2. Award the item to the bidder k with the highest priority level. That is, set the winner to be

   $$k = \operatorname*{argmax}_i \gamma_i(v_i)$$

   where γ_i is the *priority function* for bidder i. The auctioneer picks the priority functions so as to maximize its profit and announces them before the auction begins. All the priority functions must be monotone increasing (so that a higher bid gives a higher priority).
3. The price that the winner pays the auctioneer is v_k^{min}, the minimum amount that k would have needed to bid in order to win the auction. We can calculate v_k^{min} as follows. In order for k to win the auction, we must have, for all $i \neq k$, $\gamma_k(v_k) > \gamma_i(v_i)$. Equivalently, $v_k > \gamma_k^{-1}(\gamma_i(v_i))$. This implies that

   $$v_k^{min} = \max_{i \neq k} \gamma_k^{-1}(\gamma_i(v_i)) \qquad (1)$$

   Another way of looking at this is that from the perspective of bidder k, k wins the auction if it bids $v_k > v_k^{min}$, and if it wins it will pay v_k^{min}. Notice that v_k^{min} does not depend on k's bid (it only depends on the other bids). Also, if all the γ_i functions are the same, we get

   $$\begin{aligned} \gamma_k^{-1}(\gamma_i(v_i)) &= v_i \\ v_k^{min} &= \max_{i \neq k} v_i \end{aligned}$$

   in which case this auction is exactly equivalent to a second-price auction.
4. Small addendum: the auctioneer can also set a reserve price r before the auction begins. If none of the bidders i has $\gamma_i(v_i) > r$, then the auctioneer keeps the item. We also need to take this into account when setting the price: the actual value of v_k^{min} is

   $$v_k^{min} = \max(\gamma_k^{-1}(r), \max_{i \neq k} \gamma_k^{-1}(\gamma_i(v_i)))$$

There is a well-developed theory as to how the auctioneer should choose the γ_i functions and r in order to maximize its expected profit. But in this question you will derive the answers from first principles.

Questions:

(a) In general, in this auction scheme, it is a dominant strategy to bid truthfully. Why should this be the case? You do not need to write a proof: just name a feature of this auction that intuitively suggests that bidders will want to be truthful.

ANSWER: There are at least two acceptable answers:

- This auction is a generalization of a second-price sealed-bid auction, which we already know has the right property.
- Because the price k pays is independent of its bid v_k.

(b) Is it a Nash equilibrium for all agents to bid truthfully? Briefly explain why or why not.

ANSWER: Yes. A dominant strategy equilibrium is always a Nash equilibrium.

(c) Suppose Alice is a storekeeper selling an old rug at a garage sale, and she has just one potential buyer, Bob. To the best of Alice's knowledge, Bob is willing to spend between $1 and $4 on the rug (she thinks that Bob draws his valuation v_1 from a uniform distribution over [1,4]). The rug has no inherent value to Alice; she will just throw it away if Bob doesn't buy it.

Alice can try to apply our optimal auction design approach. Suppose that Bob's priority function γ_1 is just the identity (i.e., $\gamma_1(v_1) = v_1$). Then the auction boils down to the following: if Bob bids $v_1 > r$, he gets the rug and pays r (so that Alice's profit is r). Otherwise he loses and pays nothing (so Alice's profit is 0).

Define $\pi(r)$ to be Alice's expected profit when she chooses a particular value of r. Write a simplified formula for $\pi(r)$. The formula only needs to be valid when $1 \leq r \leq 4$. Clearly indicate your answer. We will not check your work. [Hint: Expected profit is the product of (a) the probability that the sale takes place and (b) the profit given that the sale takes place.]

ANSWER:

$$\pi(r) \quad = \quad \mathrm{profit}(r \,|\, v_1 > r)\,\mathrm{Pr}(v_1 > r) + \mathrm{profit}(r \,|\, v_1 < r)\,\mathrm{Pr}(v_1 < r) = r\frac{1}{3}(4-r) + 0 = \frac{1}{3}(4r - r^2)$$

(d) What value of r should Alice pick in order to maximize her expected profit? Clearly indicate your answer. We will not check your work.

ANSWER: Alice is always worse off picking $r < 1$ compared to $r = 1$, and she is indifferent between picking $r > 4$ and $r = 4$. This means that we can restrict our attention to the interval $[1, 4]$. Within this interval, $\pi(r)$ is a parabola that reaches its global maximum at $r = 2$, so this is Alice's best choice.

(e) An auction outcome is *Pareto optimal* if, after all the exchanges are completed, it is impossible to shuffle the items and money in such a way as to simultaneously make all of the agents strictly happier. [1] We really like auction mechanisms that are guaranteed to have a Pareto optimal outcome.

What happens when Bob has a $1 valuation for the rug? Is this a Pareto optimal outcome? Briefly explain why or why not.

ANSWER: When Bob truthfully bids $1, he is below Alice's reserve price of $2, so he doesn't get the rug. But the rug is actually worth a lot to him, and worth nothing to Alice. So everyone would be strictly better off if Bob paid Alice some amount of money (say, $0.50) and got the rug. The auction outcome is, sadly, not Pareto optimal.

[1] This definition of Pareto optimality is actually a slight simplification of the real definition; but use it for this problem.

Now suppose Alice has two potential buyers of her rug. Bob draws his valuation v_1 uniformly from $[1,4]$, and Carmen draws her valuation v_2 uniformly from $[0,1]$.

Again, Alice applies optimal auction design. In order to make the problem simpler, we will assume that she doesn't set a reserve price (although in reality, she would want to). We will try setting γ_1 and γ_2 to be the following functions:

$$\begin{aligned} \gamma_1(v_1) &= v_1 \\ \gamma_2(v_2) &= av_2 + b \end{aligned}$$

Alice will use the same procedure as before to try and calculate how to pick a and b so as to maximize her expected profit. Define $\pi(a,b)$ to be Alice's profit for a given choice of a and b. Repeating the rules of the auction design technique, bidder k wins if its bid has the highest priority $\gamma_k(v_k)$, and if it wins it pays v_k^{min}, the minimum valuation it could have bid and still won. As before, v_k^{min} is defined to be:

$$v_k^{min} = \max_{i \neq k} \gamma_k^{-1}(\gamma_i(v_i))$$

From Alice's perspective, her profit $\pi(a,b)$ is v_k^{min} for whichever bidder k wins the auction.

(f) What profit $\pi(a,b) = v_1^{min}$ will Alice receive from Bob if he wins the rug? Give a formula in terms of v_1 and v_2. Indicate your answer clearly. We will not check your work.

ANSWER:

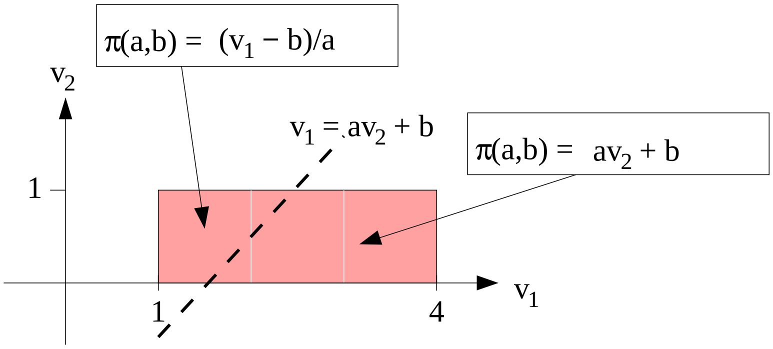
$$v_1^{min} = \gamma_1^{-1}(\gamma_2(v_2)) = av_2 + b$$

(g) What profit $\pi(a,b) = v_2^{min}$ will Alice receive from Carmen if she wins the rug? Give a formula in terms of v_1 and v_2. Indicate your answer clearly. We will not check your work.

ANSWER:

$$v_2^{min} = \gamma_2^{-1}(\gamma_1(v_1)) = \frac{v_1 - b}{a}$$

(h) The diagram below shows possible values for v_1 and v_2, which are drawn from a uniform distribution over the shaded rectangle. Fill in the values of $\pi(a, b)$ (as a function of v_1 and v_2) in the two regions divided by the dashed line.



We can use 2D integration to find the expected value of $\pi(a, b)$ and maximize with respect to a and b. But we won't make you do this during the exam. The answer is that $\pi(a, b)$ is maximized when $a = 1$ and $b = 3/2$ (and the diagram above is properly drawn to scale). Sadly, the resulting auction is not guaranteed to have a Pareto optimal outcome, as we discover below.

(i) What is the probability that Carmen will have a higher valuation for the rug than Bob does? Indicate your answer clearly. We will not check your work.

ANSWER: Probability 0. Since there is no overlap between the intervals that the two valuations are drawn from, Bob will always have a higher valuation.

(j) What is the probability that Carmen will win the rug? Indicate your answer clearly. We will not check your work. [Hint: You should be able to calculate this geometrically by looking at the area of the region in which Carmen wins in the diagram above.]

ANSWER: Carmen wins when we are in the region to the left of the dashed line in the diagram. The dashed line intersects the rectangle at $(a, 0) = (3/2, 0)$ and $(a + b, 1) = (5/2, 1)$. The region can be split into a rectangle with width 1/2 and height 1 (area 1/2), and a triangle with width 1 and height 1 (area 1/2), so overall the area of the region where Carmen wins is 1. The total area of the rectangle is 3. So the probability she wins is 1/3.