CS6601 Final – Fall 2018

Please read the following instructions thoroughly.

Fill out this PDF form and submit it on [Gradescope](). Make sure you follow the instructions on the Announcement on Canvas titled '**How to submit an exam to Gradescope**'. Remember to also submit on Canvas for a backup.

You have unlimited resubmissions until the deadline. You can: **(a)** type directly into the form – we highly recommend using Adobe Reader DC (or Master PDF on Linux). Other programs may not save your answers, so **please keep a backup**; or **(b)** print, hand-write & scan. You can combine the methods as well.

Submit only a single PDF – no phone pictures, please! (You may use an app like CamScanner or Office Lens if you do not have scanner access.) Do not add pages unless absolutely necessary; if you do, please add them at the end of the exam **only**, and clearly label **both** the extra page and the original question page. Submit **ALL** pages of the exam, not only the completed ones.

Do not forget to fill the checklist at the end before turning in the exam. The exam may not be graded if it is left blank.

The exam is open-book, open-note, open video lectures, with no time limit aside from the open period. No internet use is allowed, except for e-text versions of the textbook, this semester's CS6601 course materials, Piazza, and any links provided in the PDF itself. No resources outside this semester's 6601 class should be used. Do not discuss the exam on Piazza, Slack, or any other form of communication. If there is a question for the teaching staff, **please make it private on Piazza and tag it as Final Exam with the question number in the subject line** (for example, a question on Search would be "Final Exam #2").

Please round all your answers to 6 decimal places (unless otherwise stated). For example, 0.2345678 becomes 0.234568, 0.0012345 becomes 0.001235, 0.5672143 becomes 0.567214, and 0.1234999 becomes 0.123500. 0.12546 remains as 0.12546, and 123.23 remains 123.23. **You will not receive full credit if your answers are not given to the specified precision.**

Point breakdown *(*Each question has sub-parts with varying points):

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total
Pts	6	4	10	10	12	7	10	20	9	12	100

1. Constraint Satisfaction Problems

(6 points)

Professor Thad and head TA Theo are flying to Singapore for an AI conference. Before doing so, they need to finalize their travel plans.

They first need to stop in Korea, so they will need to find a hotel for the layover before flying once more to reach Singapore. Due to the travel agency Thad and Theo are using, they need to choose a matching hotel chain and airline for the subsequent flight as a bundle. All hotels and flights need to be bundled besides the starting flight. That is, the airline for the starting flight can be booked by itself while ignoring its bundled hotel chain, but the layover hotel and airline for the subsequent flight must be bundled together, as must the conference hotel and airline for the return flight. **Only the airline for the starting flight is exempt from the bundles**.

Use the variable names given in parentheses below to answer the questions.

There is a starting flight from Atlanta to Korea (A), a hotel overnight during the layover in Korea (B), a layover flight from Korea to Singapore (C), a hotel overnight during the conference in Singapore (D), and a flight returning from the conference in Singapore to Atlanta (E).

Use the domain names given in parentheses below to answer the questions.

The first possible airline (F1) only flies new airplanes, with new TV screens and all of the latest movies. The second airline (F2) is known for comfort, and has extra leg space for each passenger. The third airline (F3) is designed for business, and has free WiFi for its passengers. For the hotels, the first hotel (H1) has free breakfast and is bundled with F1. The second hotel (H2) has a gym and is part of a airline/hotel bundle with F2. Finally, the third hotel (H3) has the best view of the three hotels, and is bundled together with F3.

Since the start and return flights are intercontinental, Thad wants to choose an airline for at least one of those two flights that will allow him to watch some movies. Additionally, Theo needs to keep in contact with the other TAs to make some final changes to the CS6601 Final Exam, so he needs to be able to access WiFi on one of the flights on the way to the conference. Finally and perhaps most interestingly, after a bad experience in the past, Thad has a standing rule not to stay overnight in any hotel in which he might get shot at (true story).

1. After doing some research on the different hotels, Theo found out that the first hotel (H1) has an airsoft tournament on the night they would be staying during the layover in Korea (B), increasing the risk of Thad getting shot. **This constraint should also be applied for the rest of the questions.**

a) Do forward checking with the constraints given to write the domains for each variable, as well as whether each variable is arc-consistent. **(2 points)**

Variables	Domains	Arc-Consistent? (T/F)
A		
B		
C		
D		
E		

b) Is the entire network arc-consistent? **(1 point)**

- Yes
- No

2. Thad also has a standing rule not to stay in any hotel where he might see Professor Isbell. After trying to book a room at the first hotel (H1) for the night of the conference in Singapore (D), Thad and Theo were informed that the hotel chain is Professor Isbell's favorite and should thus be avoided.

a) Considering this new information, do forward checking with the constraints given to write the domains for each variable, as well as whether each variable is arc-consistent. **(2 points)**

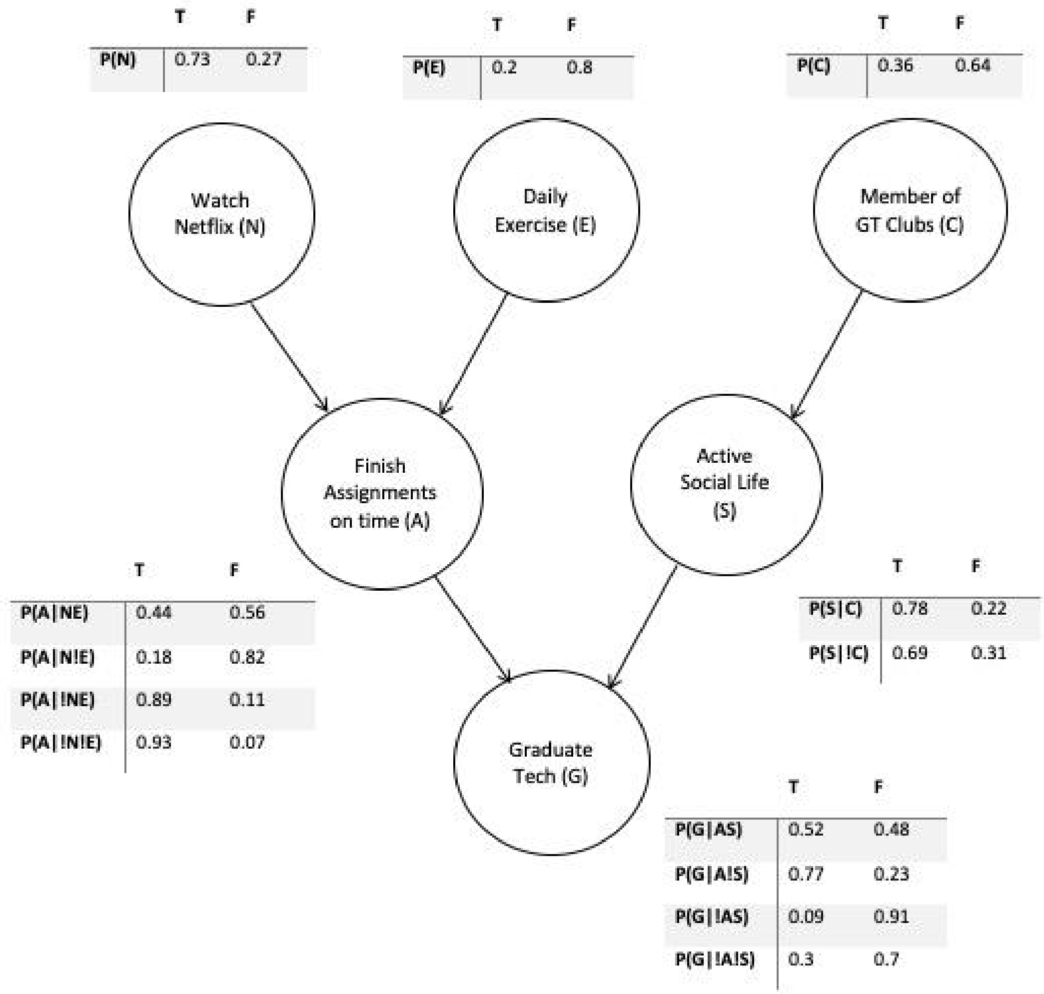
Variables	Domains	Arc-Consistent? (T/F)
A		
B		
C		
D		
E		

b) Is the entire network arc-consistent? **(1 point)**

- Yes
- No

2. Bayes Nets

(4 points)



	T	F
P(N)	0.73	0.27

	T	F
P(E)	0.2	0.8

	T	F
P(C)	0.36	0.64

	T	F
P(A|NE)	0.44	0.56
P(A|N!E)	0.18	0.82
P(A|!NE)	0.89	0.11
P(A|!N!E)	0.93	0.07

	T	F
P(S|C)	0.78	0.22
P(S|!C)	0.69	0.31

	T	F
P(G|AS)	0.52	0.48
P(G|A!S)	0.77	0.23
P(G|!AS)	0.09	0.91
P(G|!A!S)	0.3	0.7

Calculate the following probabilities using the tables and the network shown above.

P(A|E)	
P(!G|!N)	
P(C|GE)	
P(E|GN)	

3. Probability

(10 points)

In this question, we will analyze some error correcting codes. An error correcting code is a way to detect or correct errors on data transmitted via a noisy channel. Errors might happen in the transmission, so additional bits of information will be transmitted along the data, which will enable us to make some corrections.

1. **Triple modular redundancy**

The first code we will look at is the Triple modular redundancy. It simply consists on duplicating your message three times, and the receiver will take, for each bit of input, the majority vote.

For example, if we want to send the message:

0	1	1	0	1

We will send the following:

0	**1**	**1**	**0**	**1**	0	1	1	0	1	**0**	**1**	**1**	**0**	**1**

So, the recipient divides the result in three equal length tables and takes the majority vote. For example, if the second bit in the second message had an error:

0	1	1	0	1

0	**0**	1	0	1

0	1	1	0	1

Then the recipient can correct the result, and will know the message was:

0	1	1	0	1

We want to transmit a message of length 5. Let's assume each bit has a 10% chance of error during the transmission.

a. What is the probability that the 15-bit string arrives with absolutely no error? **(0.5 points)**

b. What is the probability that the 15-bit string arrives with exactly one error? **(0.5 points)**

c. What is the probability that the 15-bit string arrives with 2 or more errors? **(1 points)**

d. Let's take the first bit of the 5-bit input message. This bit will be sent three times, according to the algorithm (in the 1st, 6th and 11th position). What is the probability that this specific bit gets wrongly decoded by the recipient? **(1 points)**

e. What is the probability that the recipient gets an incorrect message, after the correction? (ie, at least one bit in the 5-bit message is wrongly decoded by the recipient) **(1 points)**

f. If you were to transmit the raw 5-bit message without any detection method, what would be the probability that the recipient gets an incorrect message? **(1 points)**

2. **The Parity Bit**

We will now look into an error detection code: the parity bit. It consists of appending one bit to the original message, containing the parity of the message. If the message contains an even number of 1s, it will contain 0, and if the message contains an odd number of 1s, it will contain 1 (this method is called *even parity bit* because the resulting array has an even number of ones).

We are not using the triple redundancy any more.

If we take again our previous example: the original message is

0	1	1	0	1

There are three 1s, so the parity bit is 1, and we transmit the 6-bit long message to the recipient:

0	1	1	0	1	**1**

The probability of a bit error (flipping from 0 to 1 or 1 to 0) in the transmission is 10% (that is valid for every bit, including the parity bit).

We are still sending a 5-bit long message (so the transmitted message will be 6-bit long).

a. What is the probability that there is at least one error during the transmission (in any bit)? **(0.5 points)**

____

b. What is the probability that the error is detected if one bit flipped during the transmission? **(1 points)**

____

c. What is the probability that the program detects the error if two bits are flipped? **(1 points)**

____

d. What is the probability that the transmitted message (including the parity bit) has some errors (at least one) and that they go undetected? **(1 points)**

____

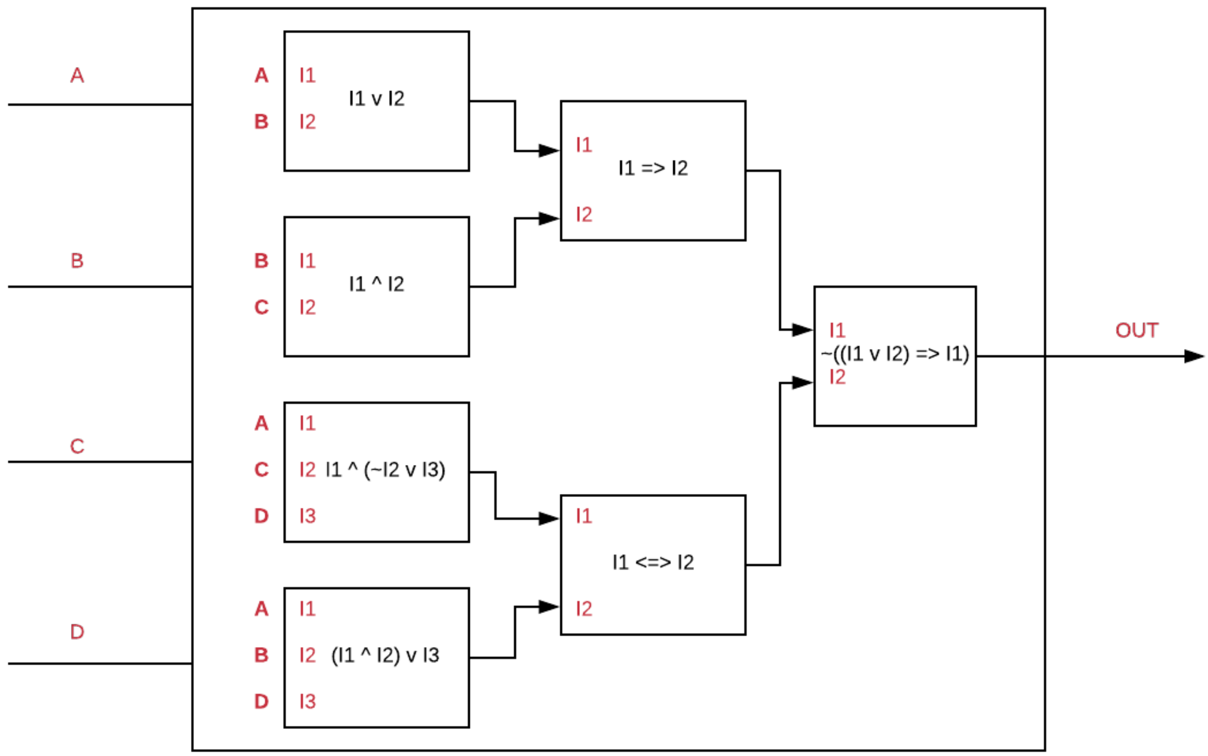
e. Let's say you want to divide your message into several blocks, every block getting its own parity bit. What is the maximum block length (not counting the parity bit) such that the probability of having an undetected error in one block is less than 0.05? **(1.5 points)**

____

4. Logic and Planning

(10 points)

1. Thad is plotting another prank on Isbell. Since the time you took your midterm, things have been quiet. This time Thad wants to play around with a chip on the Tesla's supercomputer that powers the headlights. The following schema outlines the chip's logic design.



Suppose T = True and F = False. Inputs A, B, C, & D are either T or F inputs. '~' is equivalent to 'not.'

Help Thad figure out which inputs would result in an output of False so that the headlights are never powered on.

a. Fill out the output for all possible inputs to the chip as **T or F**. **(2 points)**

A	B	C	D	OUT
F	F	F	F	
F	F	F	T	
F	F	T	F	
F	F	T	T	
F	T	F	F	
F	T	F	T	
F	T	T	F	
F	T	T	T	
T	F	F	F	
T	F	F	T	
T	F	T	F	
T	F	T	T	
T	T	F	F	
T	T	F	T	
T	T	T	F	
T	T	T	T	

b. Convert the logic in the right-most block [~((I1 v I2) => I1)] to its CNF (conjunctive normal form). Write the result below with the applicable parentheses in ascending order from left to right with the negation symbol considered less than a regular literal. **(1 points)**

____

2. A robotics student studying robot behavior at Georgia Tech observes the following behavior from Sparky The Robot.

S: (Smile => Happy) v (ThumbsUp => Happy) v (HeadBang => Happy) ^ Cool
T: (Smile ^ ThumbsUp ^ HeadBang) => Happy

After many experiments, the student concludes the following implication: **S => T**

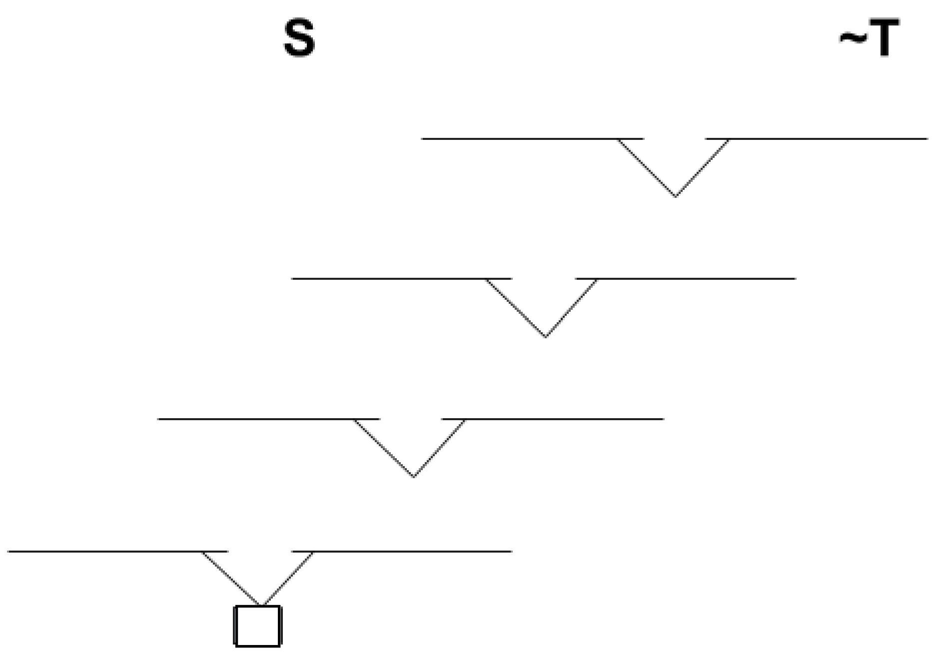
a. Convert **S** to its conjunctive normal form (CNF). Write the CNF below in alphabetical order from left to right (ex., A v B ^ ~C). **(1 points)**

b. Convert **T** to its conjunctive normal form (CNF). Write the CNF below in alphabetical order from left to right (ex., A v B ^ ~C). **(1 points)**

c. From your results in ‘a’ and ‘b’, the implication **S => T** is: **(0.5 points)**

   - ☐ Valid
   - ☐ Unsatisfiable

d. Using your results from ‘a’ and ‘b’, prove that this implication **S => T** is the choice you picked in ‘c’ by using resolution (i.e. show that **S=>T** is either valid or unsatisfiable). Fill in the resolution tree below showing your unification pairs. The tree may or may not be completely filled. **(2.4 points)**



3. The following portion of a PDDL (Planning Domain Definition Language) description represents an apple picking planning problem for a robot, which has space for storing apples.

Init(AppleField(af1) ^ AppleField(af2) ^ AppleField(af3) ^ Robot(r1) ^ Apple(a1) ^ Apple(a2) ^ Apple(a3) ^ At(r1, af1) ^ At(a1, af1) ^ At(a2, af2) ^ At(a3, af3))
Goal(In(a1, r1) ^ In(a2, r1) ^ In(a3, r1))

a. You need to add the following action to your PDDL, PutInPocket(a, r, af), which puts an apple into the robot's applicable pocket. (a=apple, r=robot, af=applefield)
PRECOND: Robot(r) ^ At(a, af) ^ At(r, af) ^ Apple(a) ^ AppleField(af)
Choose all the applicable clauses for the EFFECT **(1.2 points)**

- ☐ In(a, r)
- ☐ ~In(a, r)
- ☐ ~In(a, af)
- ☐ ~At(a, af)
- ☐ At(a, af)
- ☐ At(a, r)

b. You need to add the following action to your PDDL, Walk(r, af1, af2), which walks from applefield one's location to applefield two's location.
EFFECT: ~At(r, af1) ^ At(r, af2))
Choose all the applicable clauses for PRECOND: **(0.9 points)**

- ☐ Robot(r)
- ☐ Apple(af1)
- ☐ Apple(af2)
- ☐ At(r, af2)
- ☐ AppleField(af1)
- ☐ In(af1, r)
- ☐ In(af2, r)
- ☐ At(r, af1)
- ☐ ~In(af2, r)

5. Game Playing

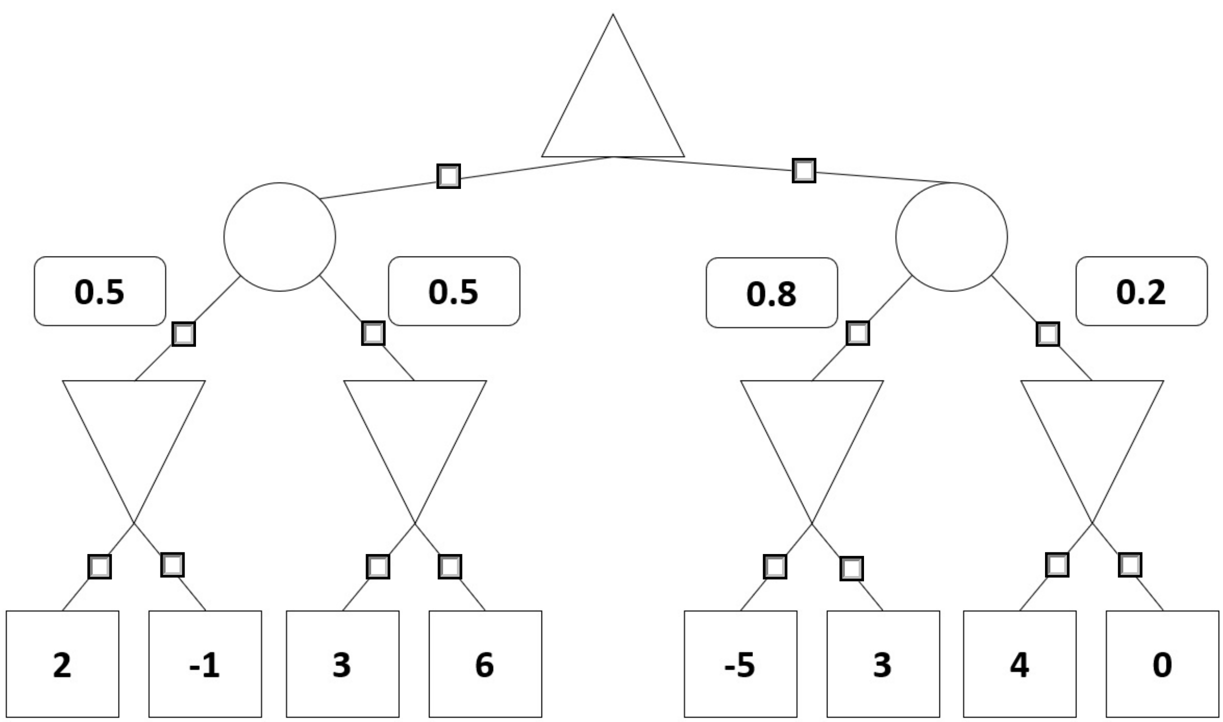
(12 points)

Question A.

(2 points)

Players A (maximizing player) and B (minimizing player) are playing a probabilistic game. The evaluation function for a given state of the game is bounded in the interval [-10, 10]. Use Expectimax with Alpha-Beta pruning to complete the game tree below. You must evaluate the tree from left to right. Check the box over branches that should be pruned, then fill in the numerical values for remaining branches. Make sure to use inequality signs where appropriate. **(2 points)**

Node types:
Max nodes = upward triangle
Minimizing nodes = downward triangle
Chance nodes = Circles



Question B.

(10 points)

The following game is for 2 players (assigned colors black and white), and the goal of the game is to construct a path of your own pieces from your side (Left or Right) to the other. The constructed path may only be a continuous path from your side of the board (Left or Right) to the opponent's side. The path may wander in any direction between the two sides of the board as long as the left and right sides eventually connect.

Game Board:

A1	A2	A3	A4	A5
B1	B2	B3	B4	B5
C1	C2	C3	C4	C5
D1	D2	D3	D4	D5

Left **Right**

Game Rules:

I. White player moves first (similar to chess).
II. Players play in turn placing one piece for each move (until the squares are exhausted).
III. Each player can only place a piece on an empty square of the board, and the piece can be of 2 types: Unpinned (U) or Pinned (P).
IV. **An Unpinned (U) piece is of the player's own color** and can be placed in any empty square.
V. **A Pinned (P) piece is always of the opponent's color**. The pinned piece flips all adjacent pieces around itself when placed. A pinned piece cannot flip neighboring pinned pieces. An example is given in Figure 2 below.
VI. The player first to complete a contiguous path in their own colored pieces from one side to the other wins. If both players form contiguous paths at the same move then it is a stalemate.
VII. If there is no path formed on the board with all squares exhausted, then the game ends in a stalemate.

In the following example game we can see that black pieces have formed a contiguous path from one side to the other thereby winning the game.

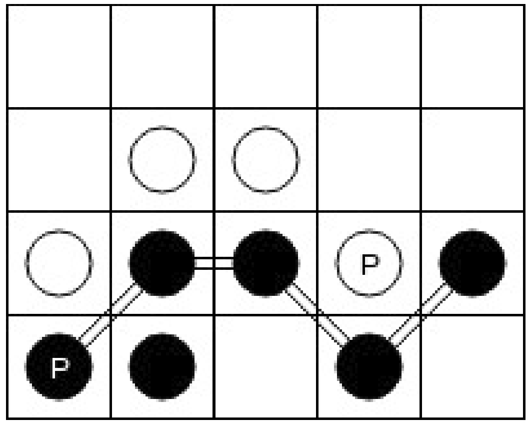


Fig 1: Example board where the Black Player won with the path shown.

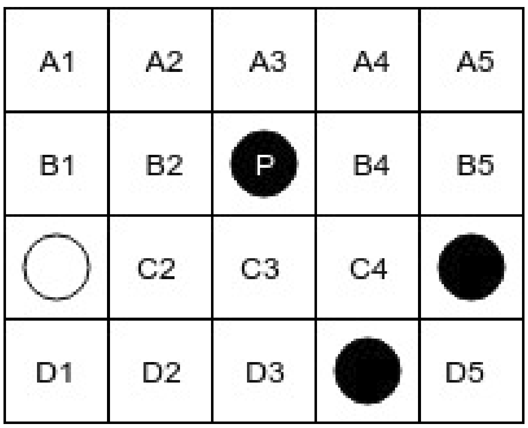


Fig 2a: Example board. Two moves have been played by both players. White's turn.

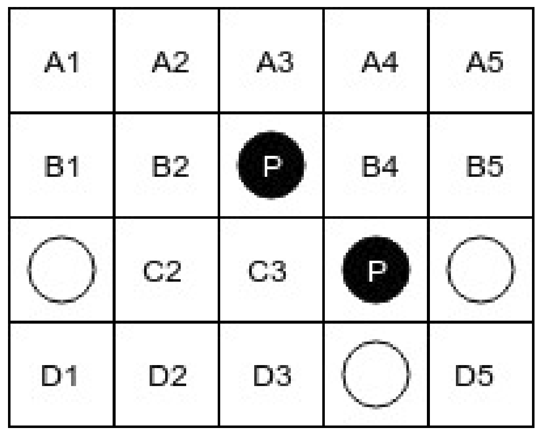


Fig 2b: Board after white player placed a pinned piece at C4.

In the example above, the White played a pinned piece (of opponent's color = black) at C4 and thereby flipped surrounding pieces at D4 and C5 but not the pinned piece at B3.
Note: the above moves are just for illustration purposes and may be sub-optimal.

1. Estimate the number of nodes in the game tree of the above game in big-O notation. Game board dimensions = h x l **(1.5 points)**

   - $O(hl)$
   - $O((hl)^2)$
   - $O((hl)^{hl})$
   - $O((hl)^{2hl})$
   - $O((2hl)^{hl})$

Assume that the two players are playing the game in a smaller version of the board (3 units x 4 units), and have reached the following stage:

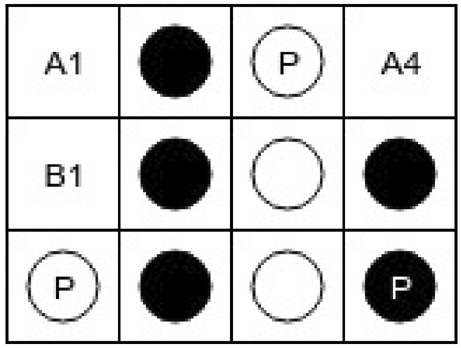


Fig 3: Game board for Q2

The evaluation function you are using is as follows:

h(.) = (max number of columns spanned by a contiguous path of my color-
max number of columns spanned by a contiguous path of my opponent's color)
+ 0.5*(number of pieces of my color - number of pieces of the opponent's color)

The first terms refer to the maximum number of columns spanned by a contiguous path for a given player. Each player may have several disconnected paths in the board at a time. The heuristic finds the one contiguous path which connects pieces of his color and spans the most columns. We then use that path to find the maximum number of columns spanned.

Example 1: If the longest chains were 3 pieces of white color (pinned or unpinned does not matter) in cells (A1 - B2 - C1) and 3 pieces of black color (B2 - B3 - C3), then the value of this part of the heuristic is 3 for player white and 2 for player black as vertical links do not count as a connection across a column. Hence, the first term ⇒ 2 - 3 = -1 if the current player playing the move is black.
Example 2: This heuristic evaluates to 1 connection at a maximum for both the black and white players in the board given above in Fig 3. Hence, the first term of the evaluation function at this stage in the game would be 1 - 1 = 0 for either players. (Note, this will be updated as the board changes)

The evaluation value h(.) for a win is 10, 0 for a stalemate and -10 for a loss.

The agent can only search two levels deep (my player making a move followed by the opponent making a move) as the branching factor is high. You may assume that the tree is evaluated alphabetically (e.g. A1, A2, A3, A4, B1, B2, B3 ...) with moves involving pinned pieces evaluated prior to that involving unpinned pieces. (e.g. PA1, A1, PA2, A2, ...)

2. Using alpha beta with depth 2, the black player decides to play a/an: **(2.5 points)**

- Pinned Piece
- Unpinned Piece

At:

- A1
- A4
- B1

3. Assume the white player also uses the same alpha-beta algorithm (depth = 2). Continuing the game, the white player will play a/an: **(2 points)**

- ○ Pinned Piece
- ○ Unpinned Piece

At:

- ○ A1
- ○ A4
- ○ B1

4. State whether the following statements are True or False:

a. Node ordering would not be beneficial to the AI agent in this case (Question B) as it will only increase the time consumed by the agent to make decisions. **(0.5 points)**
   - ○ True
   - ○ False

b. Iterative deepening can mitigate the problems raised by the Horizon effect in exploring game trees. **(0.5 points)**
   - ○ True
   - ○ False

c. Focusing on achieving secondary objectives that are more easily attainable instead of the primary objective of winning the game guarantees the agent to perform better than if it was only focused towards the primary objective. **(1 points)**
   - ○ True
   - ○ False

d. If our player using MINIMAX to play optimally and searches to end game but the opponent plays suboptimally, the value achieved by our player will always be equal to or better than if the opponent plays optimally. **(1 points)**
   - ○ True
   - ○ False

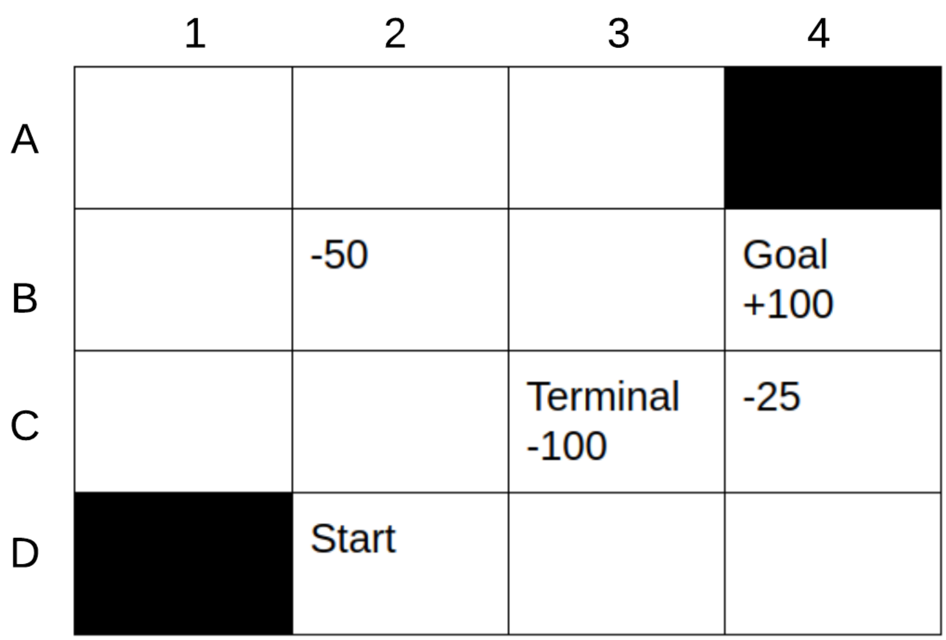
e. If our player uses MAXN instead, to play optimally in a 2 player game where the rewards are not zero-sum, and it searches to end game, but the opponents play suboptimally, the value achieved by our player will always be equal to or better than if the opponents play optimally. **(1 points)**
   - ○ True
   - ○ False

6. Planning under Uncertainty

(7 points)

A new vibranium (https://en.wikipedia.org/wiki/Vibranium) deposit was discovered in the border between Mt. Starn and Lake Isbell. The value of vibranium has been skyrocketing due to recent technological advancements that allow the use of vibranium in all sorts of previously unimaginable applications such as flying cars, shrink rays, and indestructible shields. Due to border disputes between Democratic Republic of Starneria and Autocratic Dictatorship of Isbelland, gathering the vibranium at the deposit is a dangerous task. The President of Starneria has tasked you to use a robot to mine and retrieve the vibranium from this new deposit.

After initial reconnaissance, you have enough information to model the mission as a Markov decision process, described below:



The robot starts at the state labeled “Start” (state D2). There are 2 terminal states (the Goal at B4 and state at C3). There are impassable terrain states (D1 and A4) which the robot cannot occupy. The robot can move horizontally and vertically, but not diagonally. The robot’s actions are non-deterministic or stochastic. When selecting an action, it has an 80% probability of moving in the intended direction, with 10% probability for either one of the perpendicular directions (e.g. choosing north results in 80% probability of going north, with 10% each for going west and east). If an action would move the robot toward a wall or impassable terrain, assume it stays in its current state (e.g. choosing south from “Start” would result 80 + 10 = 90% probability of staying in D2 and 10% probability of moving to D3). Assume a discount factor of 0.9. Assume a base reward -1 for states not labeled explicitly. Assume the utility of all states start at 0 (iteration 0).

1. First, you try planning using value iteration. Please fill out the table below, rounding your answers to 6 decimal places. **(2.4 points)**

State	Utility after 1 iteration	Utility after 2 iterations	Utility after 3 iterations
A3			
B2			
C4			
D2			

2. Which of the following statements are True with regards to the MDP described above: **(1 points)**

- ☐ After 3 iterations of value iteration, the policy derived from those utilities recommends going north at the starting state D2.
- ☐ Using different discount factors can result in policies that recommend different actions for a given state.

The spies of Autocratic Dictatorship of Isbelland have found out about Starneria's mission to gather vibranium at the new deposit. Isbelland, unable to deploy its own vibranium mining mission in time, reaches out to Wakanda for help. While negotiating a joint effort between Wakanda and Isbelland, The Dictator of Isbelland orders his spies to sabotage and delay Starneria's mission.

The Isbelland spies are successful: they manage to hack and delete the information gathered during Starneria's reconnaissance, as well as the plan formed using the MDP. Getting desperate, you decide to just deploy the mining robot without a plan, and have it learn as it goes using Q-learning.

3. For each transition given below, calculate the resulting Q value after the transition: a transition (s, a, r, s') means that the robot was in state s, took action a and got reward r, and resulted in state s'. Assume all Q-values initially start at 0. Assume the discount factor γ is 0.9 and the learning rate α is 0.5. **(3.6 points)**

Use the equation given here:

$$Q[s,a] = Q[s,a] + \alpha(r + \gamma max_{a'}(Q[s',a']) - Q[s,a])$$

Transition	Q[state, action]	Value
(D2, North, -1, C2)	Q[D2, North]	
(C2, North, -50, B2)	Q[C2, North]	
(B2, East, -1, B3)	Q[B2, East]	
(B3, South, -50, B2)	Q[B3, South]	
(B2, East, -1, B3)	Q[B2, East]	
(B3, South, -100, C3)	Q[B3, South]	

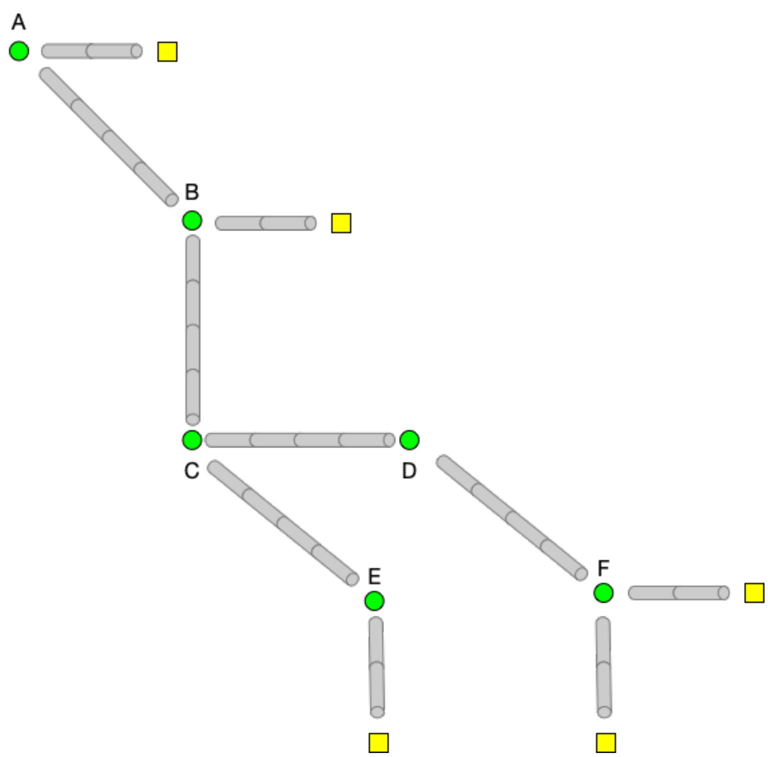
7. Search

(10 points)

Question A.

(7 points)

It’s the year 2020 and the most Boring company has just finished it’s single layer tunnel infrastructure under LA, and is about to start operations. They are seeking engineers to help them build a routing system for people, cars and cargo transportation. You are an ambitious GT graduate who passed the screening process and is now attending the first interview, where you are given the following problem to solve (refer to the map below throughout the whole Question A).



- An electric skate is a platform on wheels propelled by electric motors. They are used to carry packages from station to station.
- There are two types of stations:
    i. Company Operated - a green circle on the map, which has a capacity for any amount of skates or packages to be present at the same time.
    ii. Privately Owned - a yellow square, has a place for only one package
- Each tunnel has only a single pair of rails, meaning that only one skate can pass through it at a time. However, two skates cannot go through it at the same time (in a single time step).
- There could be multiple skates at a station at the same time. And multiple skates could be arriving at the same station at the same time from different neighboring stations.
- The environment is changing in a step by step manner, during which the skates can perform a single action (e.g. move to a neighbor state).
- The goal state is to deliver all the packages (to the privately-owned companies, that is, the yellow squares) and then bring all the skates back to their starting positions.
- At each timestep, each skate must do one of the following steps:
    i. Move to a neighboring station (don't forget that two skates can't use the same tunnels concurrently) OR
    ii. Load a package (if it currently has none) OR
    iii. Drop the package (except at a privately-owned station if the station already has a package in it)
- When all of the packages are delivered and a skate returns to the starting point (station B) it is parked there (meaning that there are no available actions it can take) .

You start with 3 skates and 5 packages at station B. The packages have to be delivered to the privately owned (yellow square) stations. The packages are the same, any of packages can be delivered to any of the destinations. Electric skates can only carry one package at a time. The skates are identical (do not identify/enumerate them individually).

Given this problem please answer the questions below:

1. Does the branching factor stay the same throughout the state space search tree expansion? **(1 points)**
    - ○ Yes
    - ○ No

2. What is the number of nodes in the first level of the tree if you are using BFS? **(1 points)**

    ____

3. Is it beneficial to use Bi-directional search (e.g. Bi-UCS)? **(1 points)**
    - ○ Yes
    - ○ No

4. If you were to use Bi-directional search would the first level of the tree be the same for both forward and backward states? **(1 points)**

   - ○ Yes
   - ○ No

5. On average, which algorithm should be the fastest to find a solution? (even if sub-optimal) **(1 points)**

   - ○ DFS
   - ○ BFS
   - ○ UCS
   - ○ A* - given an inadmissible heuristic
   - ○ Bi-UCS

6. Which of the algorithms are guaranteed to find an optimal solution (least # of steps)? Pick all the that apply: **(1 points)**

   - ☐ Iterative Deepening
   - ☐ BFS
   - ☐ UCS
   - ☐ A* - given an admissible heuristic
   - ☐ A* - given an inadmissible heuristic
   - ☐ Bi-UCS

7. Will any of the algorithms listed above (question 6) fail to terminate? **(1 points)**

   - ○ Yes
   - ○ No

Question B.

(3 points)

For the list of problems/environments/worlds given below, you have to identify what are the reasons classical search algorithms could not be applied.

1. The Tower of Hanoi puzzle in virtual reality - coded up as a bug-free computer environment - https://en.wikipedia.org/wiki/Tower_of_Hanoi
2. A robot playing the Tower of Hanoi puzzle. - Now you have the Hanoi tower in the real world, but you are executing the actions by a robot arm which not very accurate and has a success rate of 99.98%.
3. The game of Battleship (the paper version) - http://www.papg.com/show?1TMC
4. Poker - https://en.wikipedia.org/wiki/Poker
5. GO (game) - https://en.wikipedia.org/wiki/Go_(game)
6. DOTA 2 - https://en.wikipedia.org/wiki/Dota_2
7. Aircraft navigation

Here is a list of possible problems

a. The set of the actions is unknown
b. Not fully observable
c. Is not a directed graph
d. Is not static
e. Is not an MDP
f. The actions cannot be discretized
g. The states are not discrete
h. Is not deterministic
i. Is not simulatable

Which of the conditions mentioned above make the classical search algorithms inapplicable to the problems above? Please select between 0 and 3 problems with each situation.

	a	b	c	d	e	f	g	h	i
1									
2									
3									
4									
5									
6									
7									

You should only select the conditions that are relevant to the violation of the search algorithm applicability, meaning that some of the conditions could be marked as true about the problem, but if it is not relevant to search algorithms applicability you should not select them.

If you would like to provide some additional reasoning for your answers above, do it here:

---

8. Machine Learning

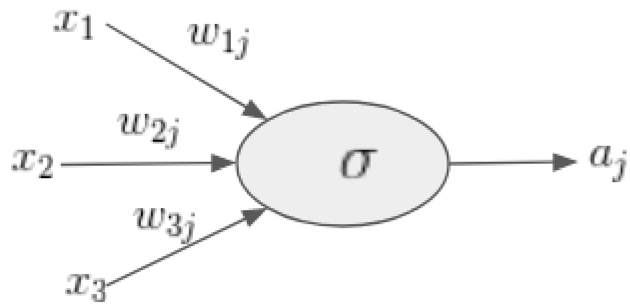
(20 points)

Question A.

(10 points)

This question mainly deals with weight updates via backpropagation in neural networks. Here, we want to walk you through the process of performing a forward pass, given a training example, computing the error and performing 1 backpropagation update to the weights.

Let us first take a look at a simple node in the network which uses the sigmoid activation function:



$$a_j = \sigma(\omega_{1j}\mathrm{x}_1 + \omega_{2j}x_2 + \omega_{3j}\mathrm{x}_3)$$

where the sigmoid function is:

$$\sigma(z) = (1 + e^{-z})^{-1}$$

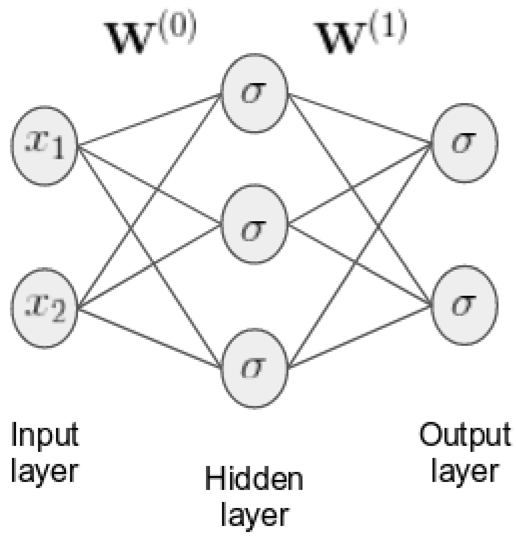
1. What is the derivative $\sigma'(z) = \frac{d\sigma(z)}{dz}$? **(0.9 points)**

- $e^{-z} * \sigma(z)$
- $\sigma(z)^2$
- $\sigma(z) * (1 - \sigma(z))$
- $\sigma(z) * (\sigma(z) - 1)$

2. What is the range of the sigmoid function? **(1 points)**

- [0, 1]
- (0, 1)
- (-1, 1)
- [-1, 1]

Consider a simple neural network with 1 hidden layer as shown below, where each edge has a corresponding parameter in the network.



Let the network parameters, inputs and outputs be represented as follows,

$$\mathbf{W}^{(0)} = \begin{bmatrix} w_{11}^{(0)} & w_{12}^{(0)} & w_{13}^{(0)} \\ w_{21}^{(0)} & w_{22}^{(0)} & w_{23}^{(0)} \end{bmatrix} \qquad \mathbf{W}^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} \\ w_{31}^{(1)} & w_{32}^{(1)} \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \widehat{\mathbf{Y}} = \begin{bmatrix} \widehat{y}_1 \\ \widehat{y}_2 \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

where $\mathbf{X}$ is the input vector, $\mathbf{A}$ is the output of the hidden layer and $\widehat{\mathbf{Y}}$ is the output of the network. Hence, the relation between $\mathbf{A}$, $\mathbf{X}$ and $\widehat{\mathbf{Y}}$ can be written as (where $\sigma(\mathbf{X})$ means the sigmoid function is applied to each element of the vector)

$$\mathbf{A} = \sigma(\mathbf{W}^{(0)\top}\mathbf{X}) \qquad \widehat{\mathbf{Y}} = \sigma(\mathbf{W}^{(1)\top}\mathbf{A})$$

Suppose we use the squared loss function, or half of the squared l_2 norm of the difference between the output and the target. This means that the following function for a training example $\mathbf{X}$ with target $\mathbf{Y} = [y_1 \ \ y_2]^\top$, needs to be minimized

$$L(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2}\left|\left|\widehat{\mathbf{Y}} - \mathbf{Y}\right|\right|_2^2 = \frac{1}{2}\left[(\widehat{y}_1 - y_1)^2 + (\widehat{y}_2 - y_2)^2\right]$$

Now that we have definitions and notations out of the way, let us do some backprop.
Consider we have the following training example pair and initial set of weights, and we wish to compute one backprop update to the weights.

$$\mathbf{X} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\mathbf{W}^{(0)} = \begin{bmatrix} 0.76 & 0.2 & 0.47 \\ 0.13 & 0.15 & 0.79 \end{bmatrix} \quad \mathbf{W}^{(1)} = \begin{bmatrix} 0.08 & 0.1 \\ 0.4 & 0.7 \\ 0.82 & 0.87 \end{bmatrix}$$

NOTE:

- There is no partial credit in the following problems, so please be careful while substituting values from previously computed answers.
- For the following set of computations round all intermediate and final computations to 6 decimal places. Meaning, after any multiplication, division, exponentiation or application of sigmoid, you need to round the answer to 6 decimal places.
- Also, you should round in the following manner:
  For values exactly halfway between rounded decimal values, round them to the nearest even value. Thus 1.5 and 2.5 round to 2, -0.5 and 0.5 round to 0, etc.

3. Compute the derivative at the output layer: **(1 points)**

$$\frac{\partial L}{\partial \widehat{y_1}} = \underline{\hspace{10cm}}$$

$$\frac{\partial L}{\partial \widehat{y_2}} = \underline{\hspace{10cm}}$$

4. Assuming the learning rate for the weights in $\mathbf{W}^{(1)}$ is 1, what are the new weights $\mathbf{W}^{(1)}$ after the update? **(1.8 points)**

$\mathbf{W}^{(1)}$ =

Now that we have updated the weights in the last layer, we continue backpropagation.

5. Compute the derivatives with respect to the output of the hidden layer. **(1.5 points)**

$$\frac{\partial L}{\partial a_1} = \underline{\hspace{10cm}}$$

$$\frac{\partial L}{\partial a_2} = \underline{\hspace{10cm}}$$

$$\frac{\partial L}{\partial a_3} = \underline{\hspace{10cm}}$$

6. Assuming the learning rate for the weights in $\mathbf{W}^{(0)}$ is 1, what are the new weights $\mathbf{W}^{(0)}$ after the update? **(1.8 points)**

$\mathbf{W}^{(0)}$ =

7. Which of the following statements are True? **(2 points)**

- ☐ Assuming you use the neural network defined right before Q2 for a learning problem on 2-dimensional data and computations are done using infinite precision, it is perfectly fine to initialize all 12 parameters ($\mathbf{W}^{(0)}$ and $\mathbf{W}^{(1)}$) in the network to 0.
- ☐ Assuming you use the neural network defined right before Q2 for a learning problem on 2-dimensional data and computations are done using infinite precision, it is perfectly fine to initialize all 12 parameters ($\mathbf{W}^{(0)}$ and $\mathbf{W}^{(1)}$) in the network to 1.

Question B.

(10 points)

1. You are a Soccer fan who is building an Information Retrieval System which returns articles relevant to Soccer out of a corpus of 1 million articles. You have come up with two algorithms which take the corpus as input and return a list of Soccer articles as output. Out of the million documents, you know that there exist only 100 Soccer articles.

Algorithm 1: Takes corpus as input and returns 100 articles out of which 90 are Soccer articles.
Algorithm 2: Takes corpus as input and returns 1000 articles out of which 100 are Soccer articles.

Metrics

I. Accuracy = $\dfrac{TruePositive + TrueNegative}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$

II. F1 score = $2 \times \dfrac{Precision * Recall}{Precision + Recall}$

a. Fill in the table with the values of the metrics obtained using the results from the algorithms above. Please round your answers to 6 decimals points. **(2 points)**

	Accuracy	**F1 Score**
Algorithm 1		
Algorithm 2		

b. Based on this evaluation which algorithm performed better at the given task of Information Retrieval and why? **(1 points)**

- ○ Algorithm 1
- ○ Algorithm 2
- ○ Both have the same performance

2. State whether the following statements are True or False with respect to Support Vector Machines: **(2 points)**

a. The kernel-trick in SVMs is primarily used to create a linear separating hyperplane in higher dimensions by converting the linearly separable original input data into a non-linear transformation.
   - ○ True
   - ○ False

b. Soft Margin Classifiers helps prevent underfitting of the training data by allowing training samples to fall on the wrong side of the decision boundary.
   - ○ True
   - ○ False

3. Assume a support vector machine with margin *d* and *k* support vectors *(k > 2).* Now we remove one of the *k* support vectors from the training set and the new margin is *d'.* What could be the possible relationships between the values of *d* and *d'*? Think of examples which satisfies the below conditions and check all that would apply: **(1.5 points)**

- ☐ $d = d'$
- ☐ $d > d'$
- ☐ $d < d'$

4. Thad has diligently collected x, y, z values from a sensor to apply to a classification task using KNN. Evil Isbell has multiplied one of the columns in the data by 1337. You are Thad's friend and want to minimize the code changes he has to make by only changing the distance metric used in his code. Which distance metric would produce the most similar results to the original? **(1.5 points)**

- ○ Euclidean distance
- ○ Manhattan distance
- ○ Minkowski distance
- ○ Hamming distance
- ○ Mahalanobis distance

5. Assume a real-valued input and one binary output (0 or 1) with a KNN (k-nearest neighbor) learning algorithm to predict Y for an input X using Euclidean distance (unweighted).You are given the dataset below:

X	**Y**
-0.4	0
1	1
1.5	1
1.6	0
4	1
4.5	0
5.1	1
5.2	0

What is the leave one out cross validation error (LOOCV) of 1-NN on this dataset? Give your answer in misclassification error percentage (accurate to two decimal places) (10% implies 10% misclassification error). **(2 points)**

____

9. Pattern Recognition Through Time

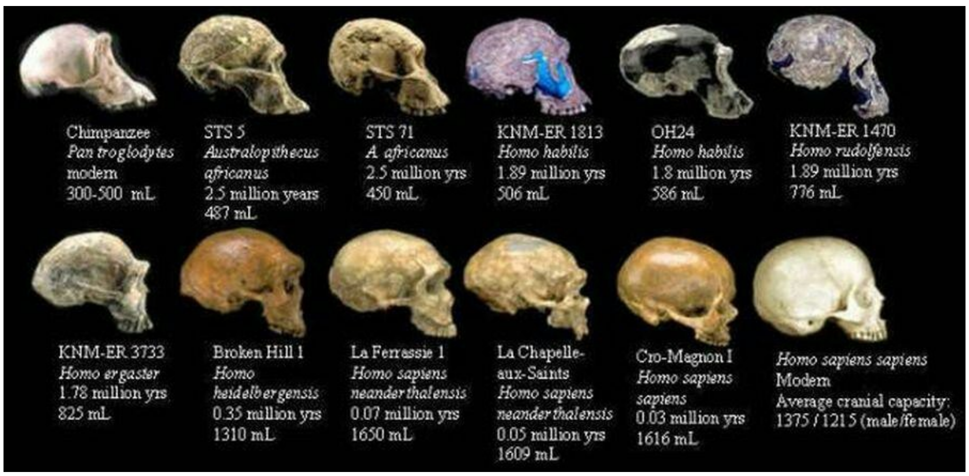
(9 points)

Question A.

(5 points)

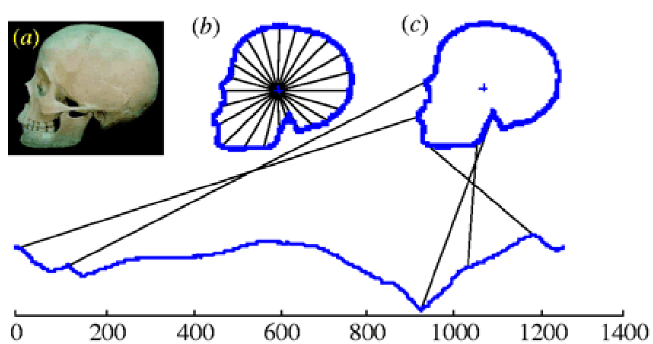
In this problem, you are going to learn how to convert images into time series for shape classification purposes.

Imagine you are given a hominid skull and you are called to classify it as one of a number of different hominid skull categories. How would you quantify the similarity?

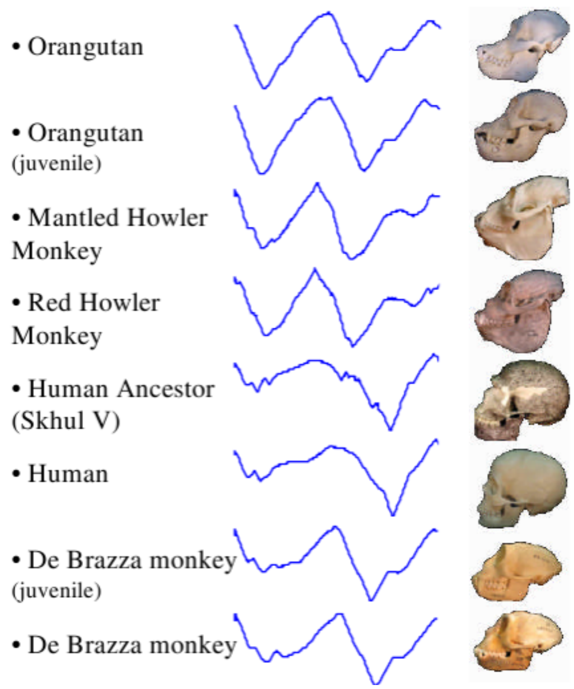


One way to approach this is to create distance measures based on time series analysis, and then perform Dynamic Time Warping (DTW), one of the most widely used algorithms that measures similarities between two time series. DTW is mostly used in tasks like speech recognition, handwriting recognition, etc., and here we are demonstrating that it can also be applied to other tasks, like shape recognition.

The first question we need to answer is: how can we convert a skull contour into a time series? We first detect the skull's outline, then we measure the distance between the center of the skull and each point on the skull's outline (b). Plotting those distance entries at regular degree intervals will ultimately give us a time series (c). In this case, we started at the skull's mouth and went in a clockwise direction around it.

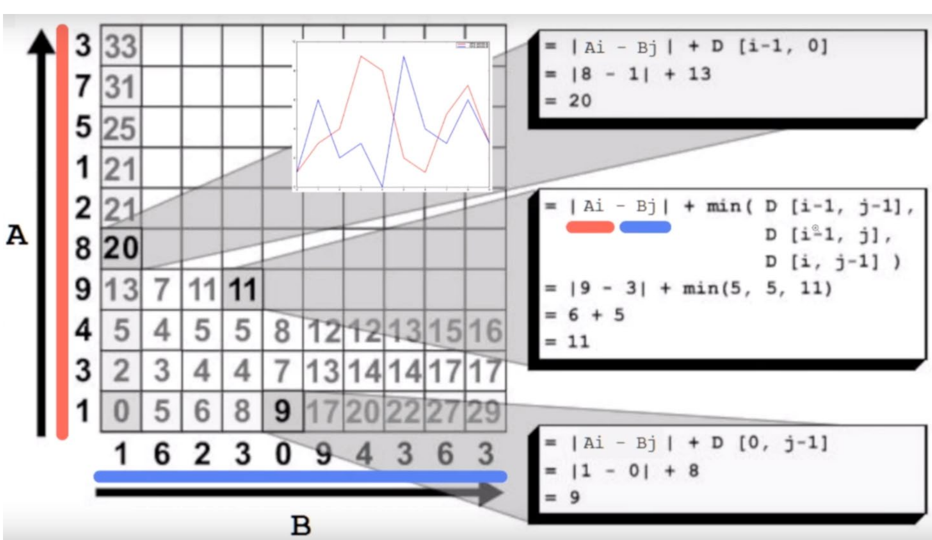


Since skulls from different species produce different time series, we can leverage that principle to determine which species a newly unearthed fossil could be, or to group similar species.

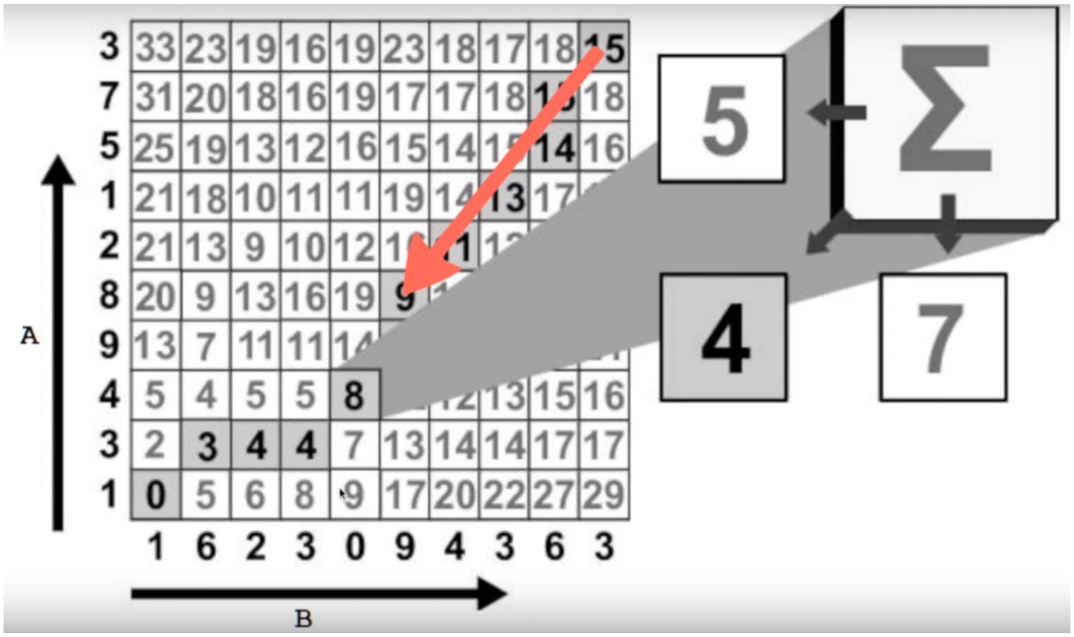


How is DTW computed? Every possible warping route between the two-time series is a path through the matrix. We want the one with the lowest final distance that connects the bottom-left square to the top-right square. Each square $\gamma(i,j)$in the grid is computed by the following formula:

$$\gamma(i,j) = diff(q_i, c_j) + min(\gamma(i-1,j), \gamma(i,j-1), \gamma(i-1,j-1))$$



When determining the shortest path along the matrix, start from the end point (top-right) and moves towards the starting point (bottom-left). At each step, you can move diagonally, left, or down. **In the case of a tie with the same values, you should move diagonally.**

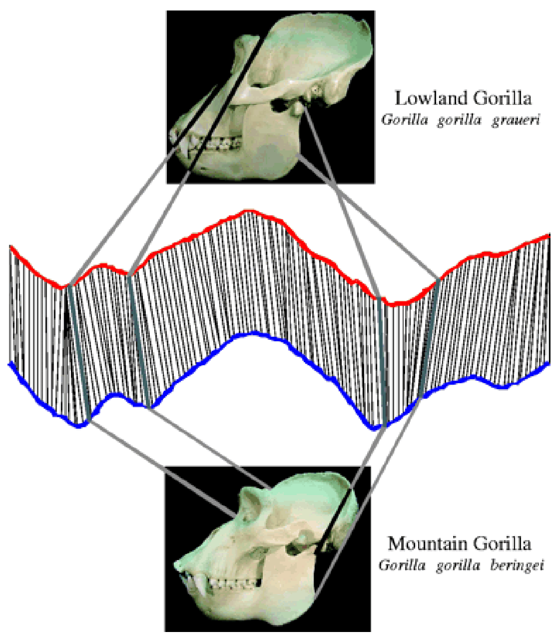


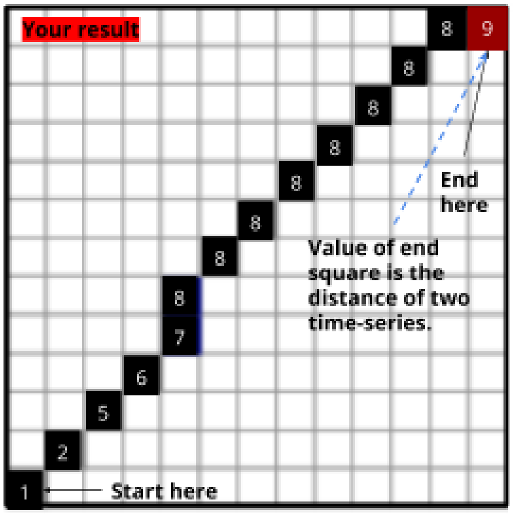
Compute the distance between the two following time series and draw the lowest route on the grid.

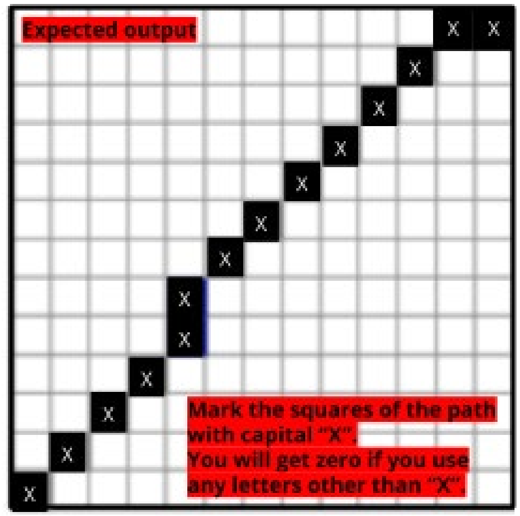
Time series of Lowland Gorilla:
[3, 2, 4, 3, 5, 6, 7, 9, 8, 5, 3, 2, 1, 5, 5, 6]

Time series of Mountain Gorilla:
[3, 2, 1, 3, 2, 5, 6, 7, 8, 6, 2, 1, 2, 4, 3, 5]

The “distance” of two time series is the value of top-right square. The lowest route should start from the bottom-left square, and end at the top-right square. Assuming you finish the DTW grid as shown in the figure below and to the left, you should write 9 as the distance, and type in X along the warping route as shown in the right figure.







1. Distance: **(3 points)**

---

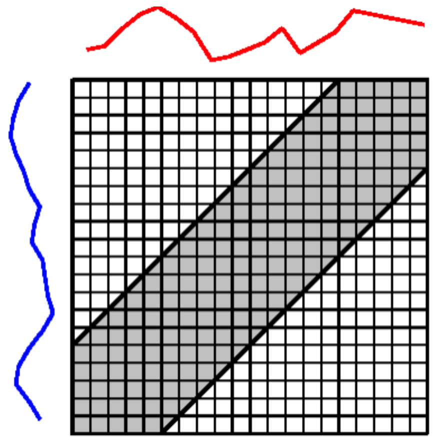
2. Lowest warping route: **(2 points)**

6																
5																
5																
1																
2																
3																
5																
8																
9																
7																
6																
5																
3																
4																
2																
3																
	3	2	1	3	2	5	6	7	8	6	2	1	2	4	3	5

Question B.

(4 points)

The naive implementation of DTW has a time complexity of $O(N \cdot M)$ and a space complexity of $O(N \cdot M)$, where N and M are the lengths of the two time sequences. This is the most cited reason to not use DTW. How can we make it faster? Based on your result from Q1, you should observe that a lot of grids, mostly grids around the bottom-right and top-left corners, were left unused. We can exploit that observation to reduce unnecessary computation. Sakoe-Chiba bounds are one way to realize that idea. In this section, you are going to compute the distance between two impulse time series with different widths of Sakoe-Chiba bounds.



Sakoe-Chiba band

Given two impulse signals:
A = [0, 0, 1, 2, 1, 0, 0, 0, 0]
B = [0, 0, 0, 0, 1, 2, 1, 0, 0]

Compute the distance between them with different widths of Sakoe-Chiba bounds. The tables below are provided for your convenience to compute the value manually. **You don't have to fill in anything in the table**. Grayed out grids are outside of the bands and you don't have to take those values into consideration.

1. Distance **(1 points)**: ____

0	--	--	--	--	--	--	--	--	
0	--	--	--	--	--	--	--		--
0	--	--	--	--	--	--		--	--
0	--	--	--	--	--		--	--	--
1	--	--	--	--		--	--	--	--
2	--	--	--		--	--	--	--	--
1	--	--		--	--	--	--	--	--
0	--		--	--	--	--	--	--	--
0		--	--	--	--	--	--	--	--
	0	0	0	0	1	2	1	0	0

2. Distance **(1.5 points)**: ____

0	--	--	--	--	--	--	--		
0	--	--	--	--	--	--			
0	--	--	--	--	--				--
0	--	--	--	--				--	--
1	--	--	--				--	--	--
2	--	--				--	--	--	--
1	--				--	--	--	--	--
0				--	--	--	--	--	--
0			--	--	--	--	--	--	--
	0	0	0	0	1	2	1	0	0

3. Distance **(1.5 points)**: ____

0									
0									
0									
0									
1									
2									
1									
0									
0									
	0	0	0	0	1	2	1	0	0

The purpose of this question is to show you how important the width of the Sakoe-Chiba bands is. When the width is zero (without any warping), you are basically computing the Euclidean distance between the two sequences. With some warping allowed, we can reduce the distance by mapping the impulse spike. However, with unlimited warping, you might end up wrongly classifying two very different samples into a same group.

10. Optimization Algorithms

(12 points)

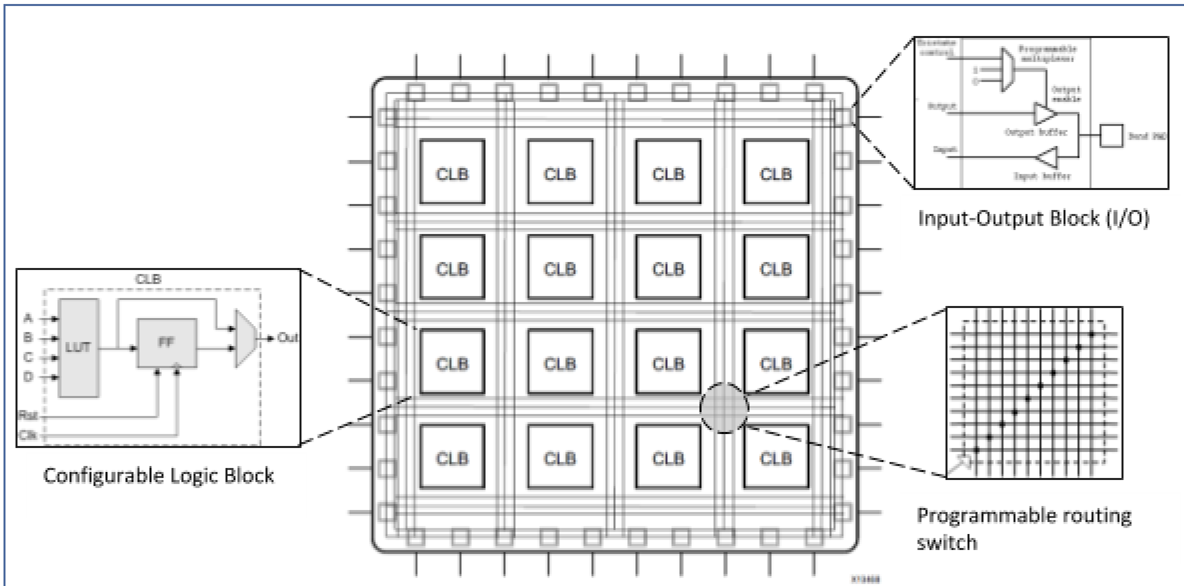
Apply random optimization to the FPGA Placement problem.

A ‘Field-programmable gate array’ (FPGA) is a pre-fabricated silicon device (IC integrated circuit) that can be electrically programmed in the field to realize almost any kind of digital circuit or system. It can be programmed as a microprocessor, a video decoder, a GPU, a crypto miner or even all of these at once. As the name implies these devices are reprogrammable, i.e an FPGA working as a microprocessor can be reprogrammed to work as a GPU in the field, as opposed to other class of silicon devices (ASICs) which are fabricated for a specific functionality.

An FPGA is made up of thousands of Configurable Logic Blocks (CLB) embedded in an ocean of programmable interconnects (routing switches). Mainly, an FPGA consists of three blocks as described below.

1. **CLBs**: Base logic block used for implementing digital logic, these blocks are made of Look-Up Tables (LUTs), Flipflops and Multiplexers, as shown in the figure below.
2. **IOB (Input-Output Blocks):** This block is used to connect the FPGA with external pins.
3. **Routing switch:** Connects a routing channel between two CLBs or between a CLB and an IOB.

Apart from CLBs, IOBs and routing interconnect, FPGAs also contains dedicated hard-silicon blocks such as Microprocessors, RAM, DSPs, External Memory Controller, PLLs, PCIe Bridge controller, etc. For this question, we will assume a basic FPGA device consists of CLBs, routing switches, and I/O blocks.



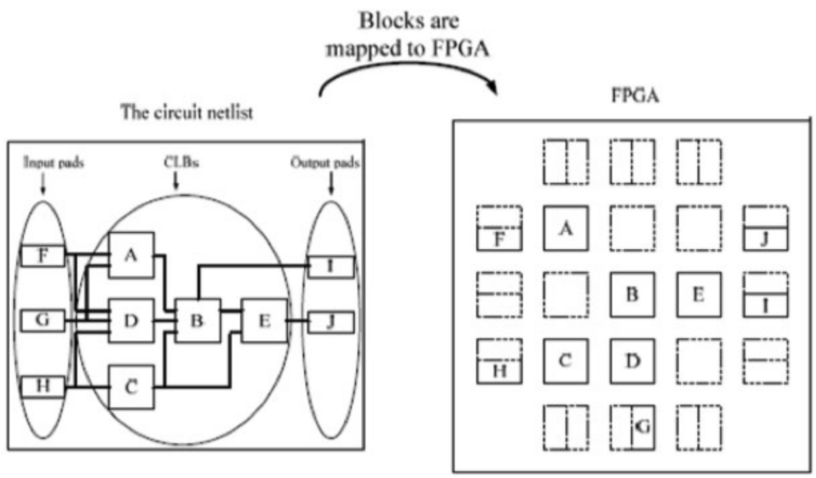
FPGA Architecture

Field-Programmable Gate Arrays (FPGAs) provide a means for fast prototyping and cost-effective design. Their reconfigurable nature, lower implementation cost and shorter time needed to realize a given design has made them a viable implementation alternative for larger, more complex designs. One of the greatest factors that determine the performance of a circuit programmed in the FPGA is the effectiveness of the placement algorithm.

FPGA placement algorithms try to place the clusters of logic, representing the digital design, onto the array of FPGA CLBs such that:

1. The critical path (the longest path from either a primary input to a primary output, a primary input to CLB, CLB to CLB, or CLB to primary output) is minimized.
2. The power consumption of the programmable routing is minimized.
3. The overall wire-length of the mapped circuit is minimized.

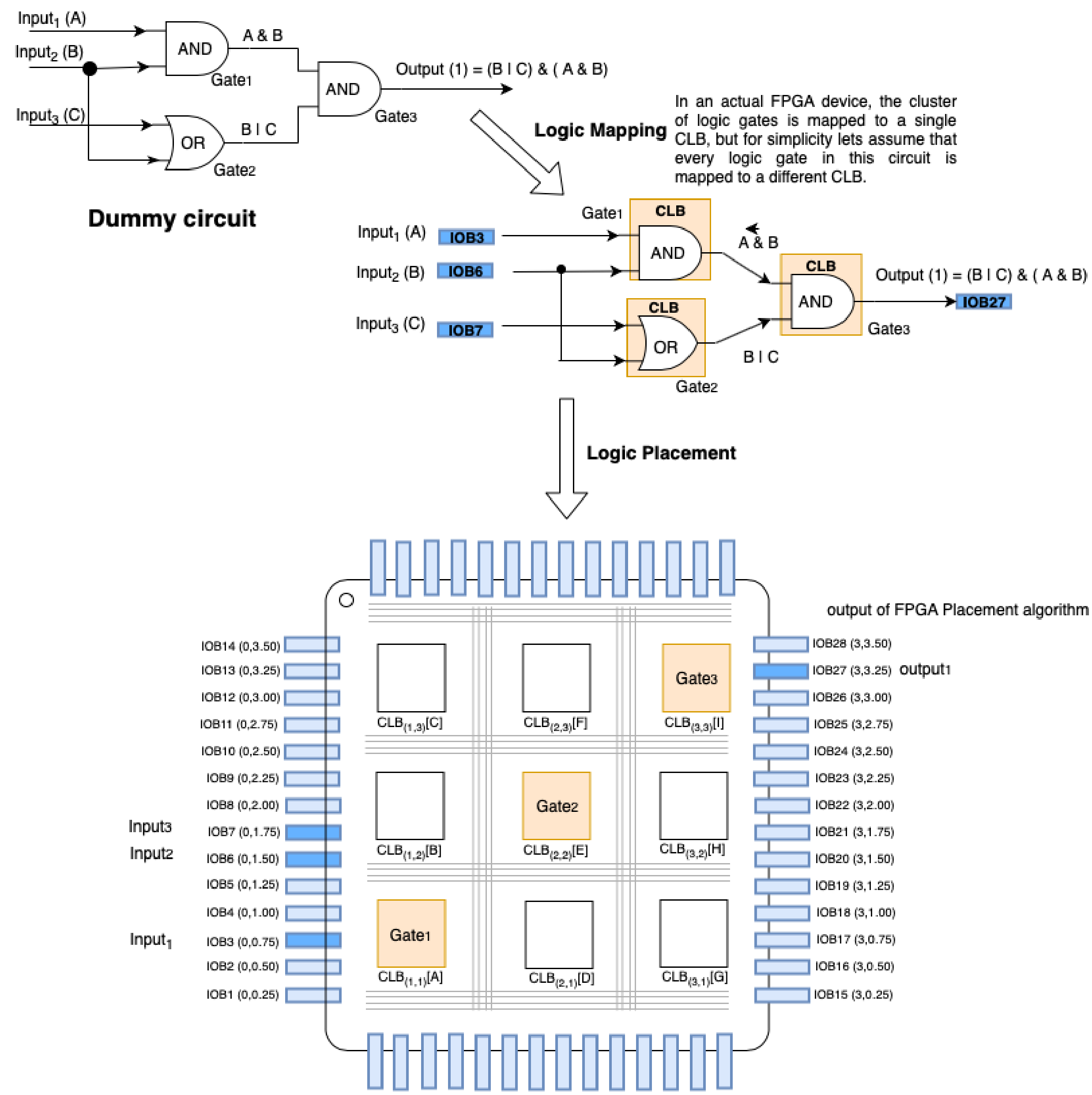
The placement algorithm places logic blocks in the network to specific positions in the FPGA, as shown in the figure below. The correct placement of logical blocks in FPGA is an optimization task, which involves optimizing multiple objectives.



In this question, you will apply optimization algorithms such as simulated annealing and genetic algorithms to find the near-optimal design placement for a given circuit.

Representing a State/Genome

Each CLB location on the FPGA array is a gene, and the 2-D location of all active CLB forms an individual genome. The figure below shows a genome for a circuit design consisting of 3 logic elements (gates).



Circuit Elements	$Input_1$	Input2	Input3	Gate1	Gate2	Gate3	output1
FPGA CLB/IOB location	(0,0.75)	(0,1.50)	(0,1.75)	(1,1)	(2,2)	(3,3)	(3,3.25)
FPGA CLB/IOB ID	3	6	7	A	E	I	27

Placement State

Due to external PCB constraints input and output IOBs are fixed, so we can ignore IOBs while representing state.

The **Placement State** here is **AEI**, in the order of the gate IDs (numbers).

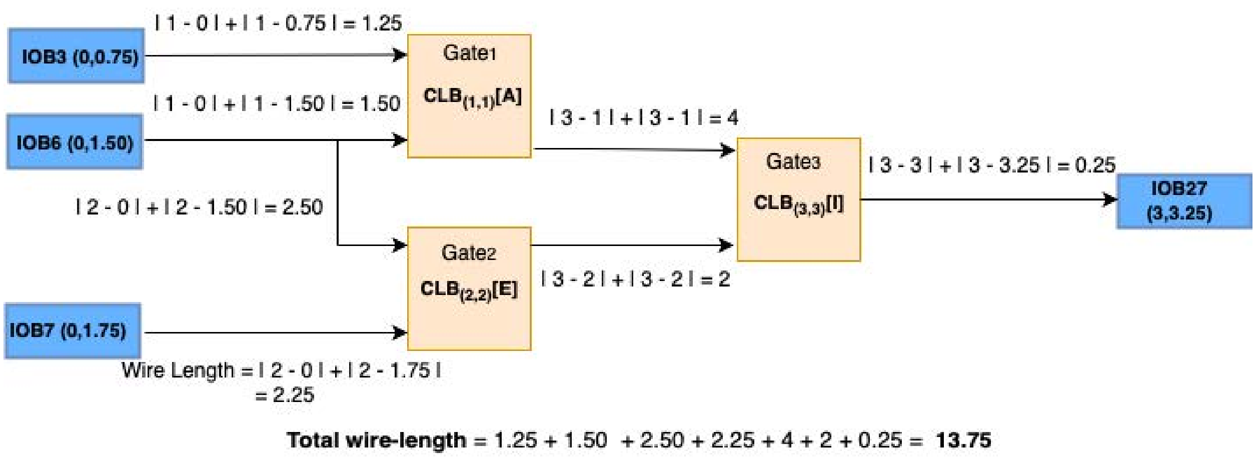
Fitness/Cost Function

Broadly, there are three placement objectives i.e. wire length-driven, routability-driven, and timing-driven placement. Wire length-driven placement tries to place logic close together to minimize the required wiring. The routing-driven placement balances the wiring density across the FPGA whereas the timing driven placement minimizes the length of a critical path to meet timing constraints. For simplicity, we will only consider the wire length driven placement objective. The total interconnection length of all the CLBs with mapped digital logic used in an FPGA can be calculated as sum of Manhattan distances between two individual locations of CLBs/IOBs as shown below. The fitness function is the inverse of total wirelength cost.

Fitness score = 1 / (Total Wirelength Cost)

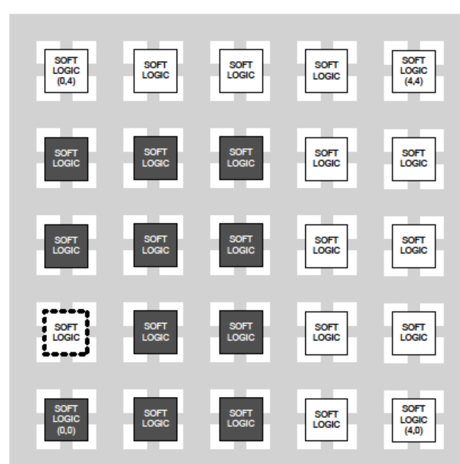
Total Wirelength Cost = $\sum_{i,j}^{N}(\left| x_i - x_j \right| + \left| y_i - y_j \right|)$, *where N is pair of active connections.*

For example, the total wire-length cost for the 'AEI' dummy circuit given above is 15.25 as shown below.



R-Limit

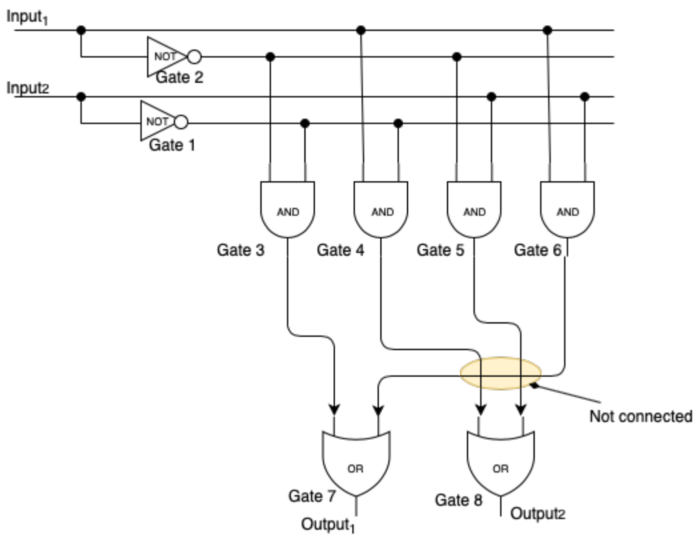
R-Limit is a constant value which determines the maximum allowable distance for swapping two CLBs. For example, if you're using an R-Limit of 2 you can only swap a certain CLB with the CLBs that are 2 cells away from it. The figure on the right shows a 5x5 FPGA where random swaps could happen for digital logic at (x, y) = (0, 1) with an R-limit = 2 (The candidate cluster to swap is surrounded by a thick dotted line and the clusters it can swap with are shaded in darker grey). The CLBs are referred to as a SOFT Logic in this figure. The figure is from "Revisiting Genetic Algorithms for the FPGA Placement Problem" by Peter Jamieson.



Question A.

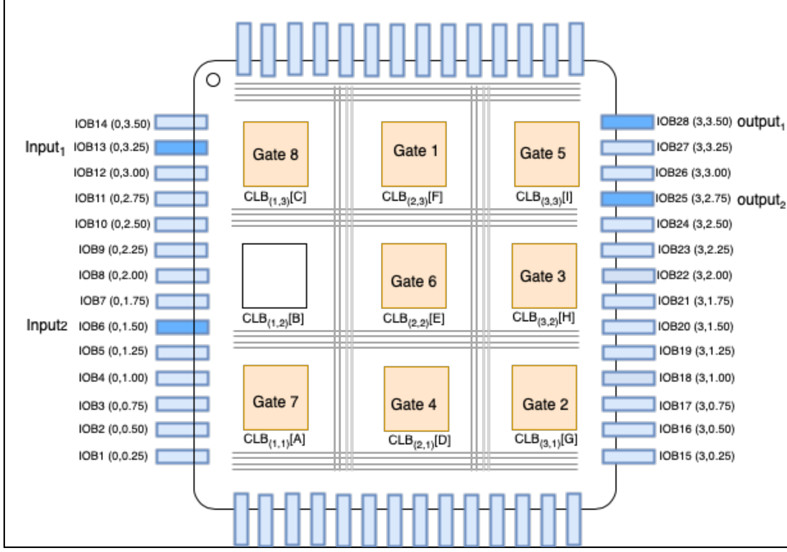
(7.5 points)

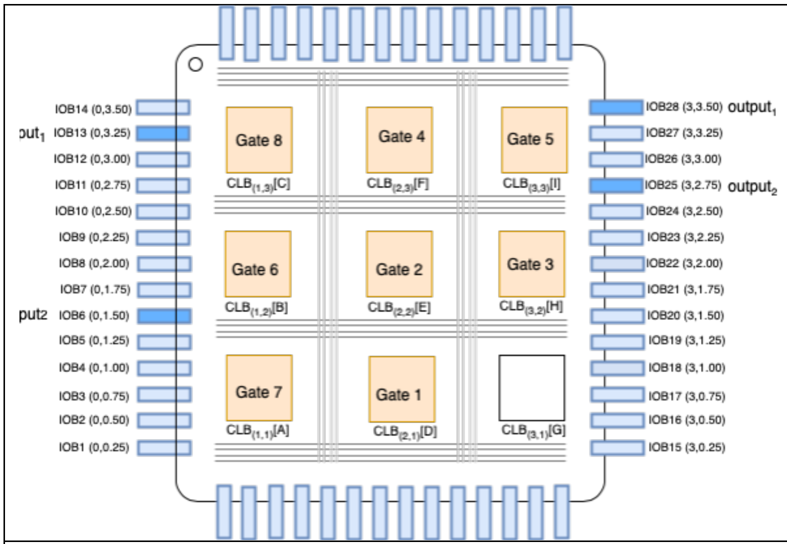
For this question you will use Genetic Algorithms to solve the FPGA placement problem. For the given circuit below use a genetic algorithm to find a near-optimal placement. Assume that each gate in this circuit will be mapped to different CLBs.

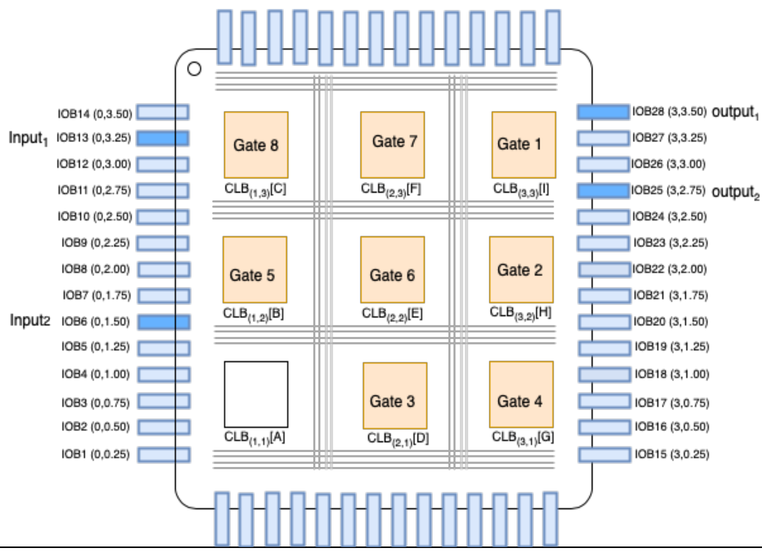


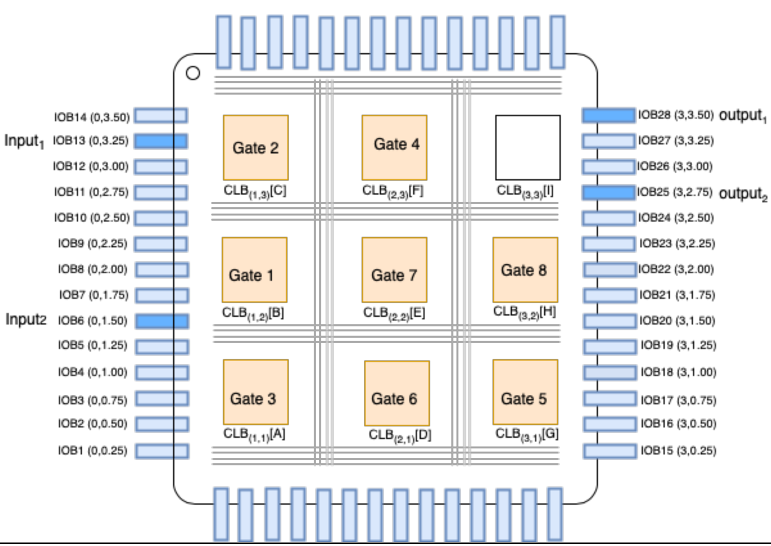
Step 1: State Representation & Fitness score calculation

1. Please write down the state, wirelength and the fitness score for each of the placements below. These four placements make up your initial population. The placement state for a) is provided as an example. **(1.5 points)**

Random Placement	Placement State	Total wire length	Fitness score
	FGHDIEAC	47	0.0212766

<table>
<tr>
<td>


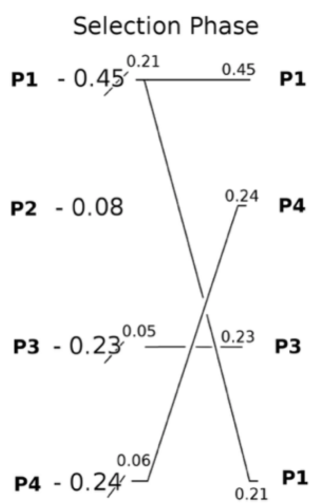
</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>


</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>


</td>
<td></td>
<td></td>
<td></td>
</tr>
</table>

Step 2: Selection criteria

You will be making a selection based on the fitness score.



Measure the proportional probabilities of the population and pick the most promising placement as your first parent (if it is a tie, take any of them). The figure on the left is an example of this procedure (but does not show the actual probabilities calculated by the fitness function).

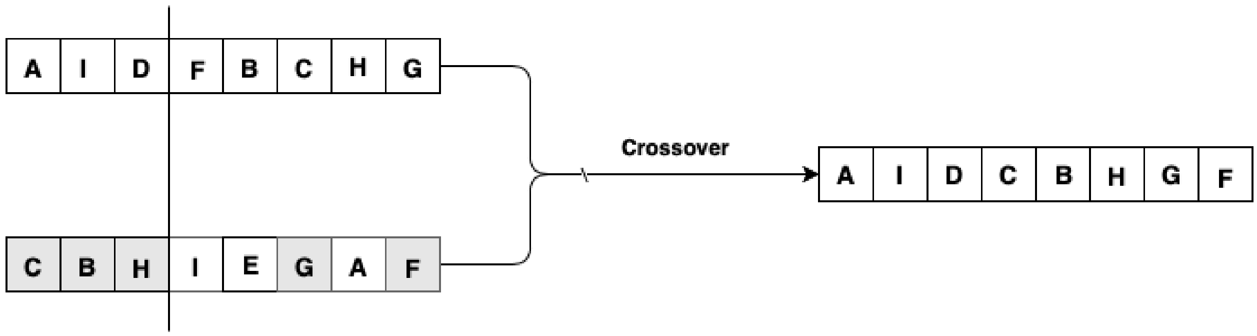
Every time you pick a genome as a parent, square its probability so that you give a chance to other genomes as well. Then select the new most promising genome as the second parent, square its probability, and keep going until all 4 parents are selected. Some genomes may be selected twice, and some may not be selected at all.

2. Please write down the placement state for the four selected parents (use the initial population given in the previous question as the initial generation). (**1 points)**

	Placement State
a)	
b)	
c)	
d)	

Step 3: Crossover

Now perform the crossover between the parents. The crossover point will be between digits 3 and 4. The normal crossover process would produce invalid sequences, where different logic gates may get mapped to the same CLB. In order to counter that, each child will get the first 3 characters from one parent and other characters of that parent will be added in the order they were found in other parent. If a character is not found in other parent, then that character will be added at the end. Here is an example:



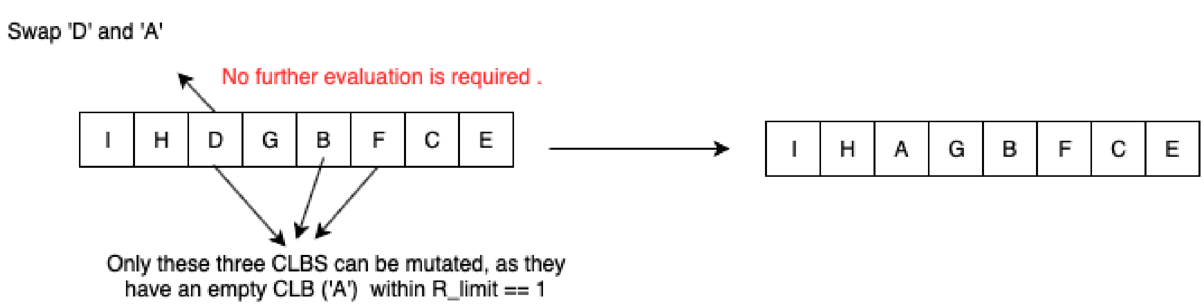
3. Please write down the placement state for the four offsprings. **(1 points)**

	State
a)	
b)	
c)	
d)	

Step 4: Mutation

Every time you pick a genome to mutate, from left to right evaluate every logic gate and find whether it is possible to swap the logic with an empty CLB within a distance of R_limit =1, if it is possible then swap CLBs, stop further evaluation as soon as one valid swap/mutation is found. Here is an example:

Mutation Example



4. Please write down the placement states for the four offsprings of Q3 after mutation. **(1 points)**

	State
a)	
b)	
c)	
d)	

5. Assume that the initial population represents generation number 1. Please write down the best two genomes at the beginning of generation number 4 (a generation changes every time you do crossover and mutation) (If there is a tie, return either.) **(3 points)**

	State
a)	
b)	

Question B.

(4.5 points)

The simulated annealing algorithm, when used in the FPGA domain for placement, attempts to minimize the various metrics for optimization based on the cooling schedule. Basically, a cooling schedule controls the weighting of a probability function. This function determines whether a randomly selected swap between the mapping of two CLBs on an FPGA is accepted or not. At each step, the logic mapped to two different CLBs are swapped. Each swap could either improve or degrade the energy function. Initially, all swaps are accepted regardless of whether they improve the optimization metric. As the temperature cools, only swaps that improve the critical path are accepted.

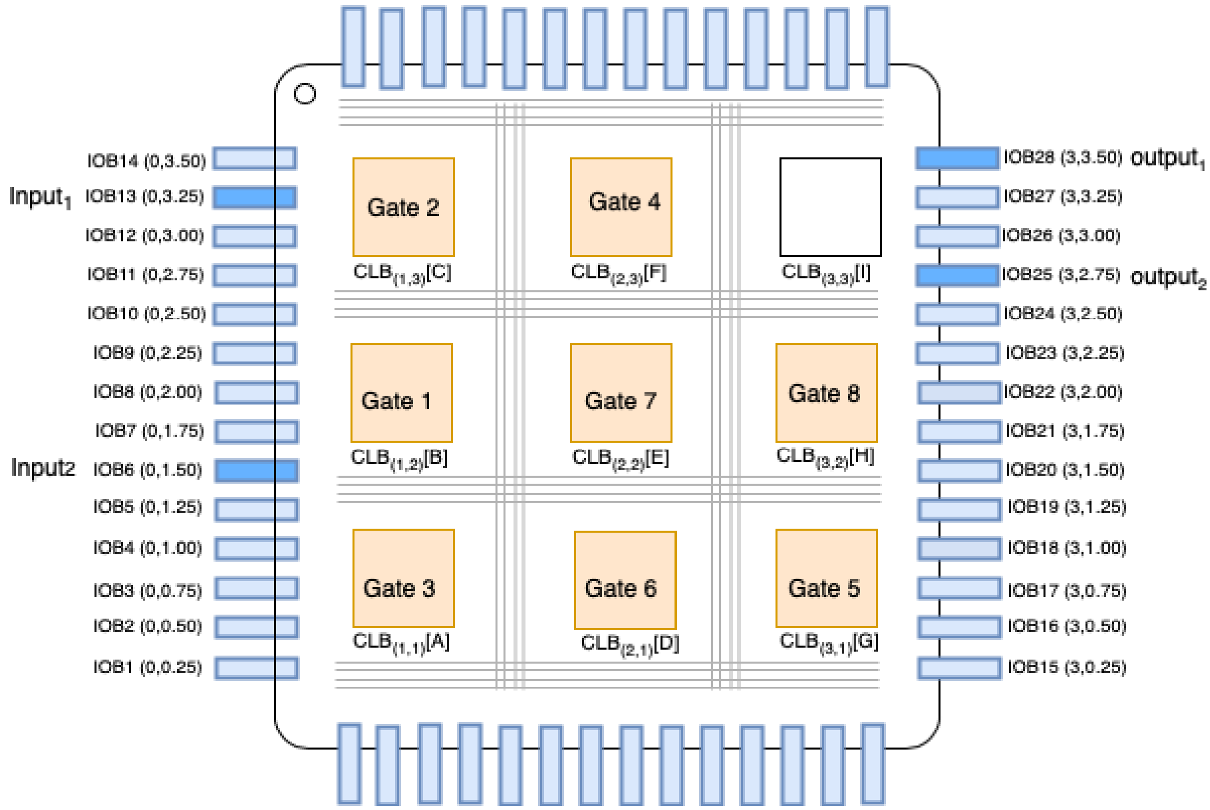
Prob (Accept) = $e^{-\delta E/T}$, if $\delta E > 0\ else\ 1$,

Where $\delta E = (TotalWireLength_{current} - TotalWireLength_{previous})$

1. Calculate the probability that a given swap will be accepted by the Simulated Annealing Algorithm. When doing the calculation for a given set (row) below, assume that the previous swap was accepted. You should use the anneal temperature from the same row for the calculation. Do not compute a joint probability. **(4 points)**

Selecting a block to Swap with: Find all possible CLBs to swap with based on the R_limit distance and choose the one in alphabetical order. Same block cannot be selected twice to be swapped with in an algorithm.

Initial Placement for a circuit given in Question 10.A is shown in the diagram below.



Step	Block to be swapped	Current placement state	Temperature	R_limit	Select a candidate to swap with (Same block cannot be selected twice)	δE	Acceptance probability
1	F	BCAFGDEH	100	2	A	0	1
2	C	BCFAGDEH	80	2			
3	I		60	1			
4	B		40	1			
5	E		20	1			

2. Is the solution found by Simulated Annealing in the question above better, worse, or of the same wirelength as the solution found by the Genetic Algorithm in question A.5? **(0.5 points)**

- ○ SA solution is better in terms of fitness score.
- ○ SA solution is worse in terms of fitness score.
- ○ Both have same wirelength.

11. Extra Credit

(Upto 2 points)

CIOS! We will be awarding extra credit to the entire class based on CIOS completion rates.

The completion rate for CIOS is defined as follows:

(number of students who completed CIOS) / (number of students enrolled in the section)

The completion rate (up to 1.0) will be multiplied by 2 and added to the score of your final exam.

Please consider taking the CIOS! As we have said over the course of the semester, this course is constantly evolving, and we are looking at ways of making it better. We take CIOS feedback very seriously and incorporate it into our teaching methods wherever possible.

Checklist

And now mark the checklist below making sure you have taken care of each of the points mentioned:

- [ ] I have read the pinned Piazza post with the title 'Final Exam Clarifications Thread', and I am familiar with all of the clarifications made by the Teaching staff.
- [ ] All answers with more that 6 digits after the decimal point have been rounded to 6 decimal places.
- [ ] All pages are being uploaded in the correct order that they were presented to you.
- [ ] Any extra pages **(including blanks)** are only attached at the END of this exam, after page 48 with clear pointers to wherever the actual answer is in the PDF (reference properly).
- [ ] Are you submitting only one PDF and nothing else (no docx, doc, etc.)?
- [ ] Is the PDF you are submitting not blank (unless you want it to be)?
- [ ] **I will go over the uploaded pictures on Gradescope and make sure that all the answers are clearly visible. I acknowledge that I am aware that dull/illegible/uneven scans will not be graded.**