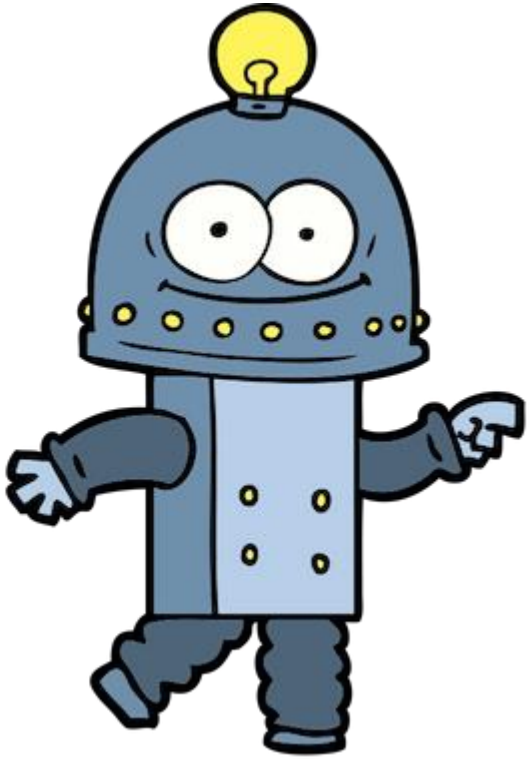
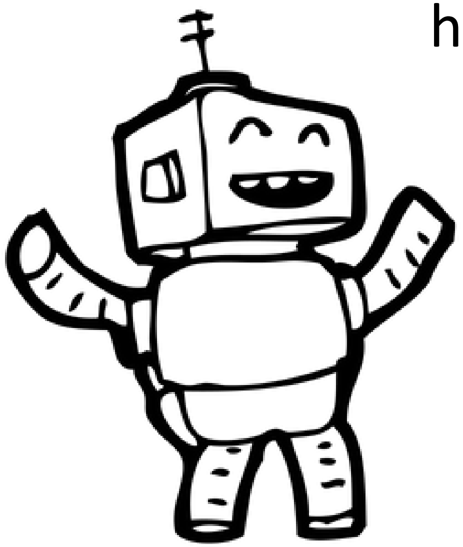


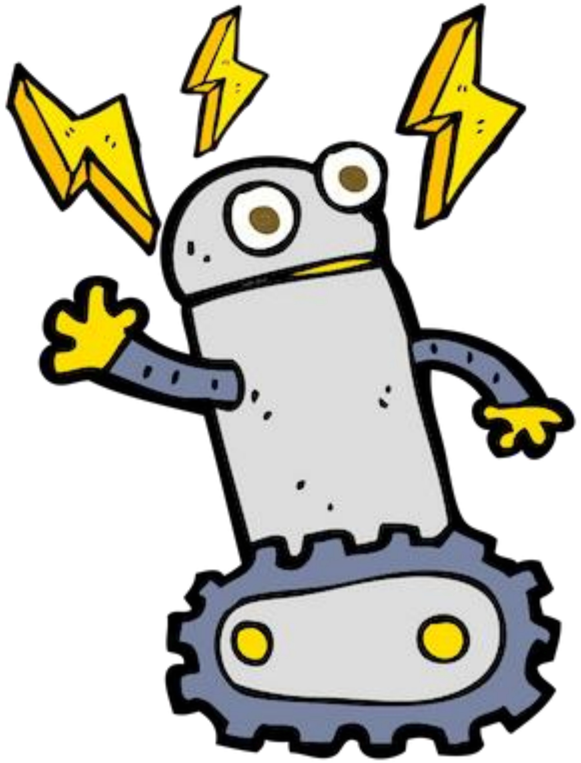
Assignment 1

Inf2D

The Assignment is out now!

https://www.inf.ed.ac.uk/teaching/courses/inf2d/coursework/

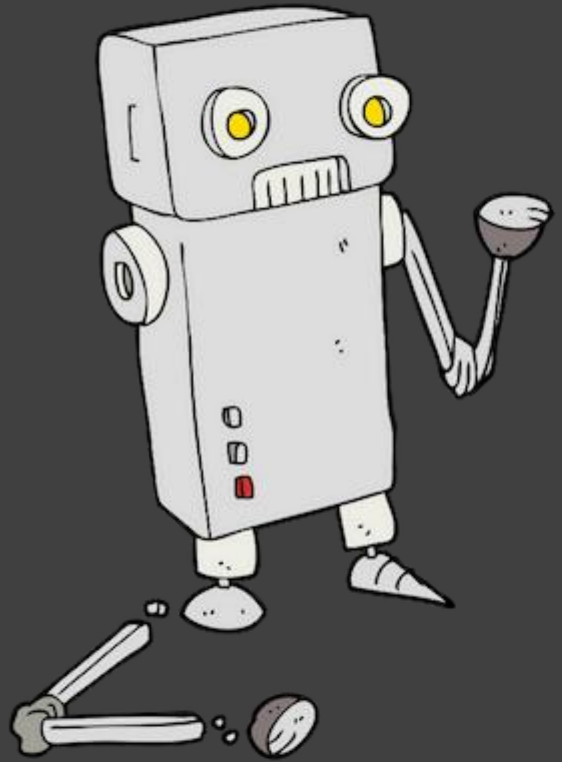




The Deadline is

3 PM, Tues,
10 March 2020

Coursework Clinics (Labs)



- Time: 12 – 2 pm
- Every Friday
- In Week: 4, 5, 6, 7, 8, and 9
- Demonstrators: Bora Alper, Ben Cottier, Raman Goyal and Alan Paul

- Get help from Demonstrators if you are stuck or have a question
- Have a regular space and time to get started on the assignment

More Help

Read R&N Chapters 3 (Search) and 5 (Games)

Piazza:

Ask and answer questions!

Email me, Stefanie Speichert (s.speichert@ed.ac.uk)

Keep an eye out on the mailing list

What is this Assignment about?

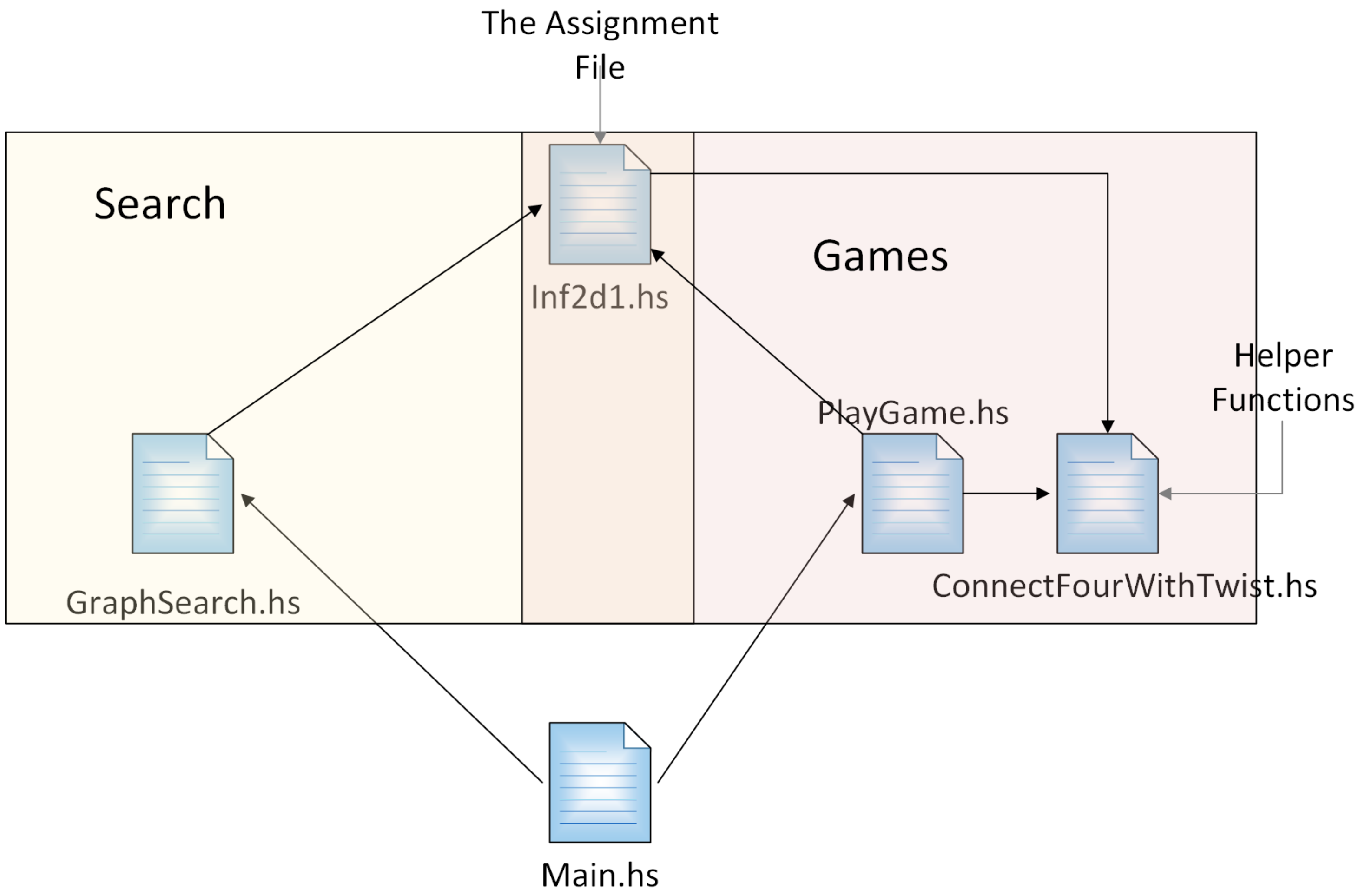
- Search
  - Uninformed Search:
    - Breadth-First Search
    - Limited-Depth Search
  - Informed Search:
    - A* Search
  - Theoretical Questions

- Games:
  - Connect Four with a Twist:
    - Minimax with Alpha/Beta Pruning
  - Quadrio:
    - Theoretical Questions

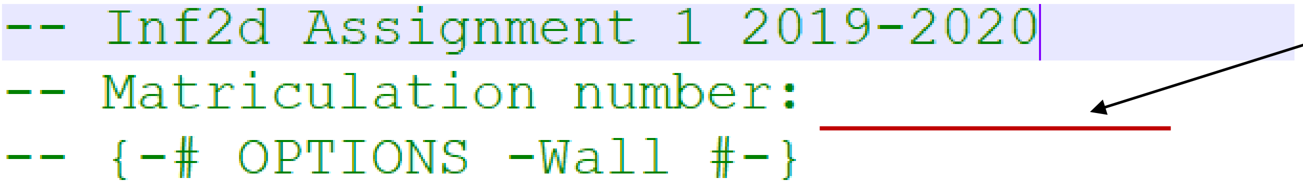
ℹ The assignment is coded in Haskell!

Assignment Layout

- 5 Files
- Inf2D1: This is your assignment file. Everything you modify will be in this file!
- For Search you will interact with GraphSearch.hs
- For Games you will interact with the PlayGames.hs file
- All helper functions for both games are in the file ConnectFourWithTwist.hs

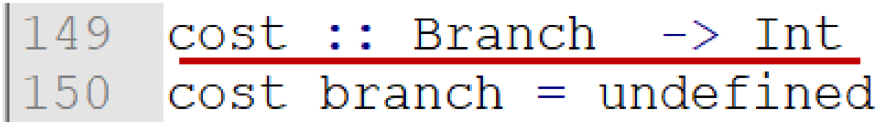


The Inf2D1 File



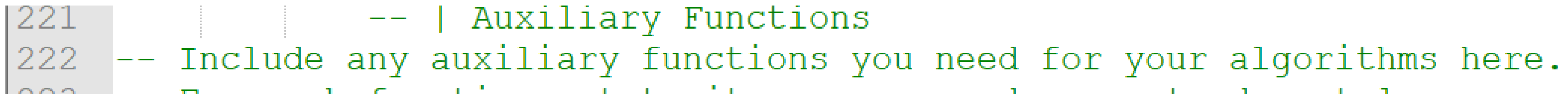
```
-- Inf2d Assignment 1 2019-2020
-- Matriculation number:
-- {-# OPTIONS -Wall #-}
```

Your Matriculation Number goes here



```haskell
cost :: Branch  -> Int
cost branch = undefined
```

Only modify the body of a function, NEVER its types and name



```
            -- | Auxiliary Functions
-- Include any auxiliary functions you need for your algorithms here.
```

All auxiliary Functions come AFTER this line

Be wary:

Test your code with all edge cases!

Your code should always finish within 2 minutes!

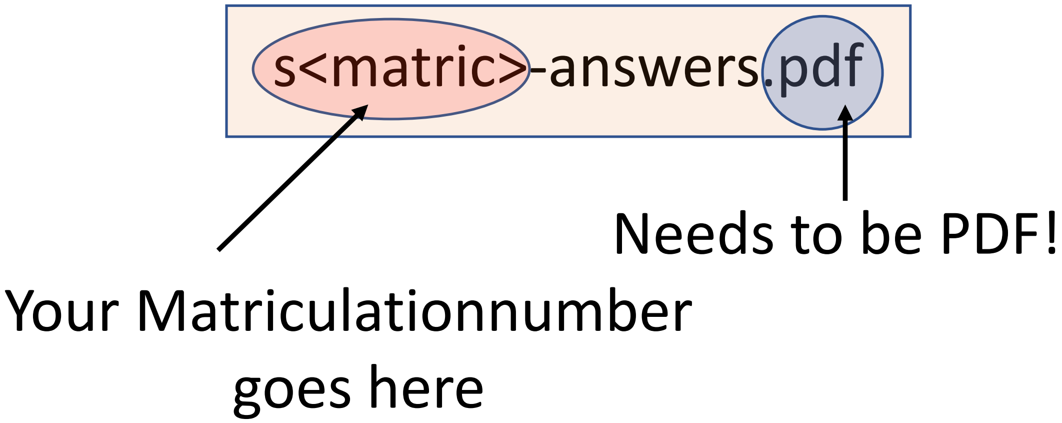
Do not modify any other file than Inf2d1!

What is graded?

- Correctness
- Efficiency (All search algorithms have to terminate in less than 2 mins)
- Code Quality: Variable Naming, Comments, ...

How to Submit: Formatting The Files

Make a PDF out of your answers and Rename it according to this format:

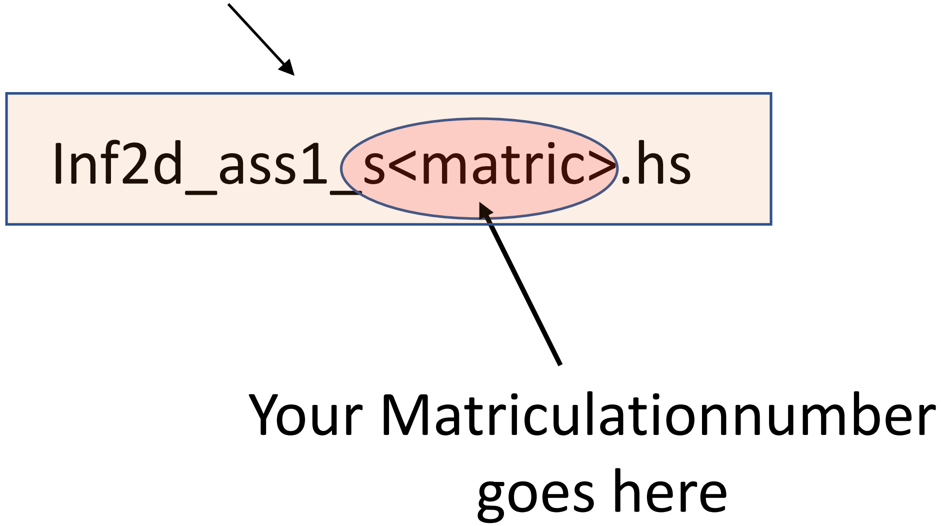


s<matric>-answers.pdf

Needs to be PDF!

Your Matriculationnumber goes here

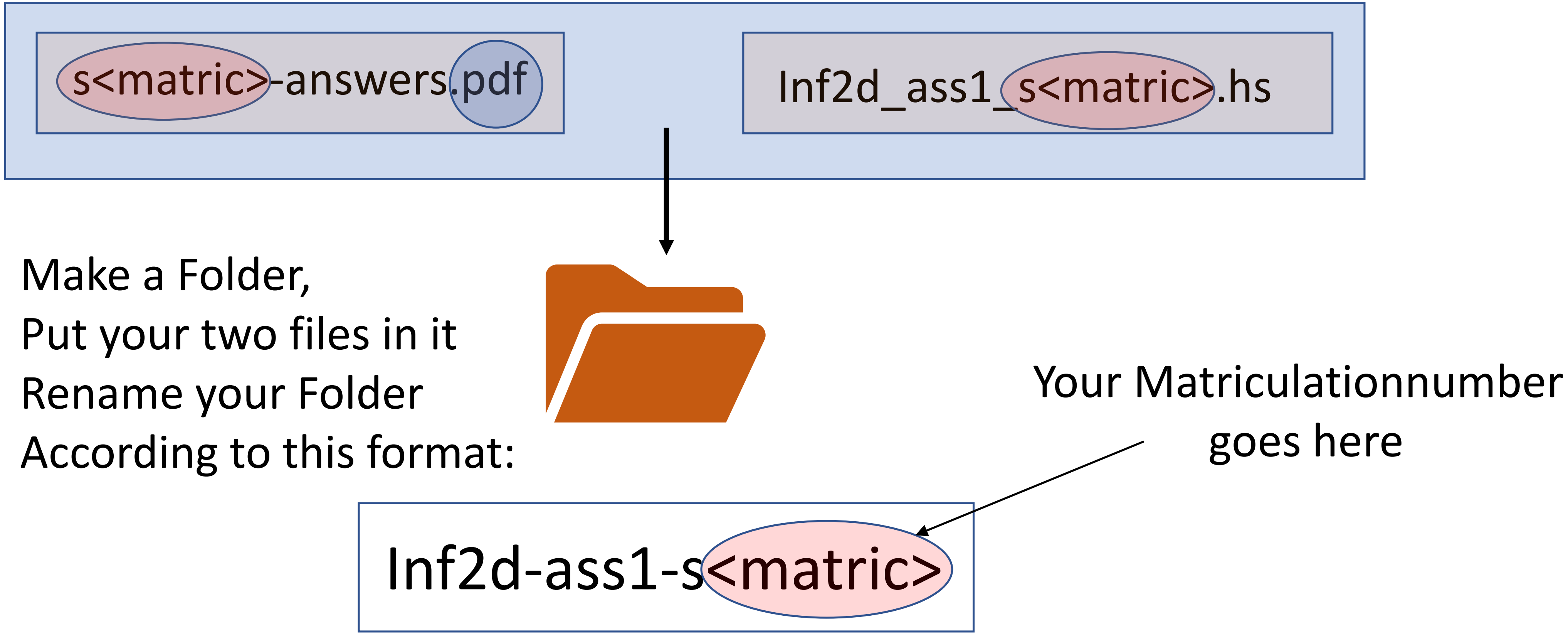
Rename Inf2d1.hs according to this format:



Inf2d_ass1_s<matric>.hs

Your Matriculationnumber goes here

How to Submit: Formatting The Files



s<matric>-answers.pdf

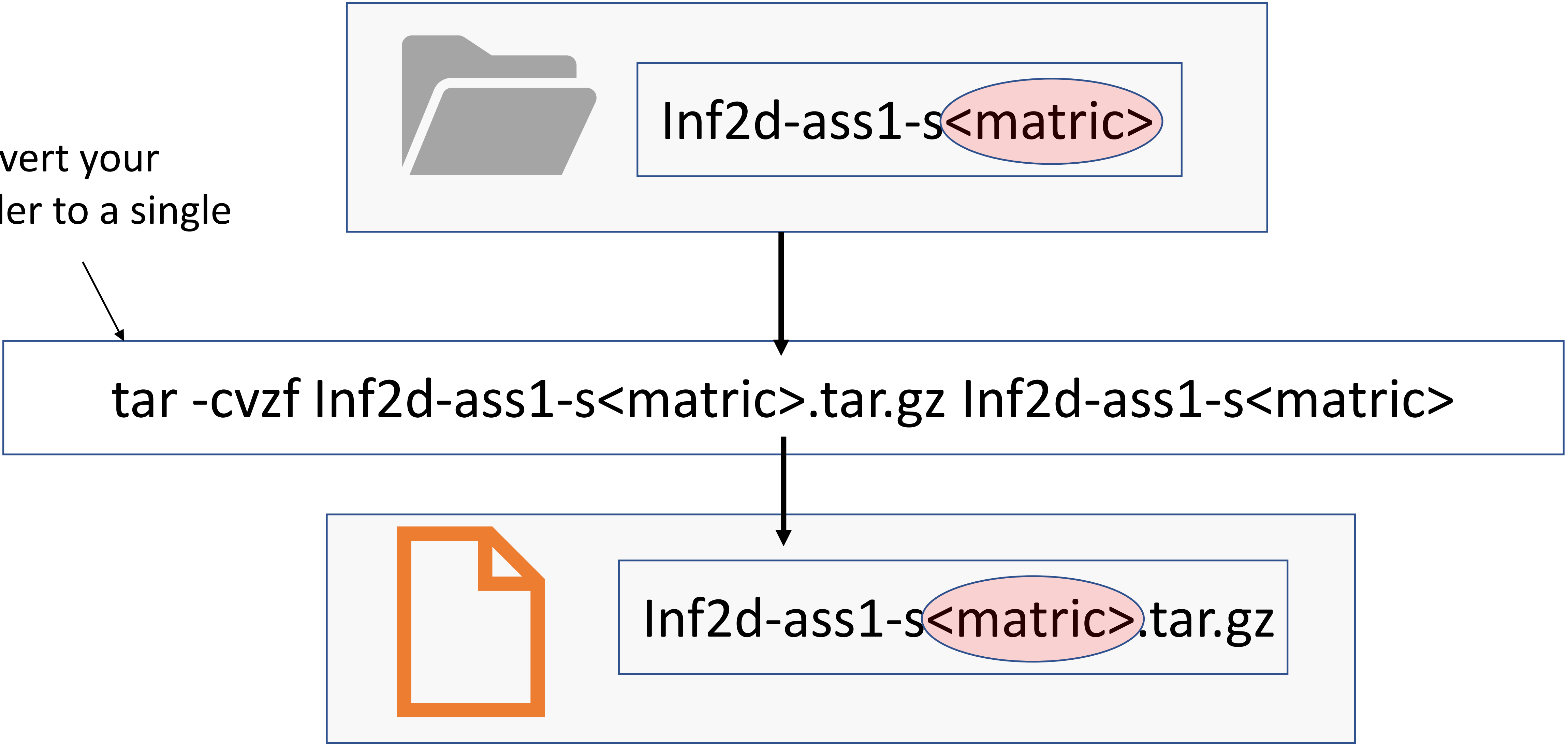
Inf2d_ass1_s<matric>.hs

Make a Folder,
Put your two files in it
Rename your Folder
According to this format:

Inf2d-ass1-s<matric>

Your Matriculationnumber
goes here

How to Submit: Formatting the Folder



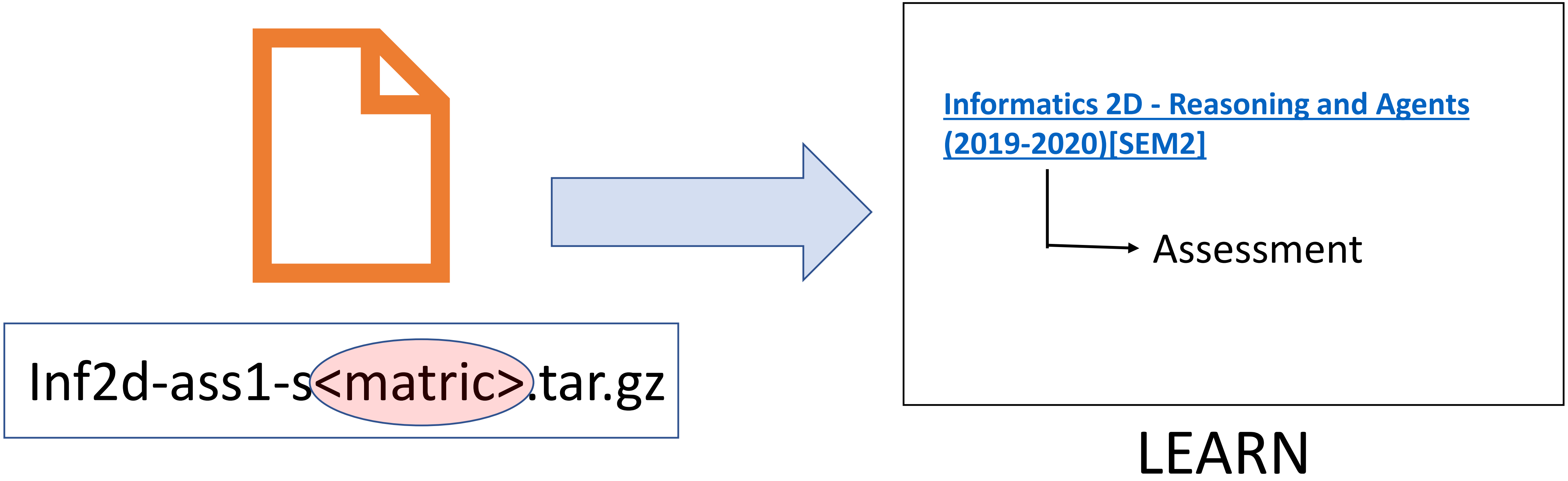
Inf2d-ass1-s<matric>

Convert your Folder to a single File

```
tar -cvzf Inf2d-ass1-s<matric>.tar.gz Inf2d-ass1-s<matric>
```

Inf2d-ass1-s<matric>.tar.gz

How to Submit: LEARN Submission



Inf2d-ass1-s<matric>.tar.gz

Informatics 2D - Reasoning and Agents (2019-2020)[SEM2]

Assessment

LEARN

You can submit any number of times until the deadline!

Extensions

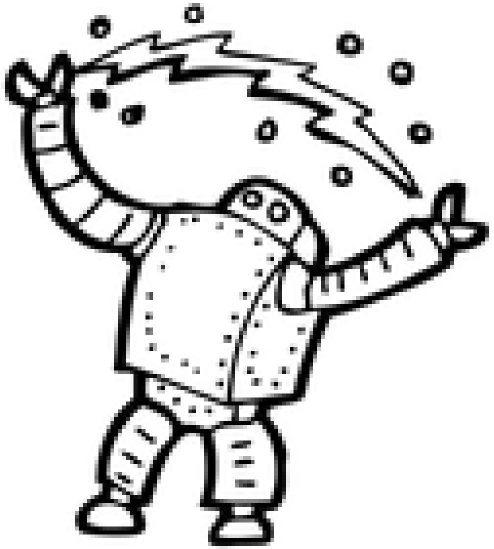
- An extension can only be given
  - If you qualify for special circumstances
  - by the School NOT the lecturer of a course. Contact ITO if you need one.
- You do not have to inform the course organiser if you are given an extension, simply submit in the timeframe you were given!
- Standard Late Submission penalties apply if you submit after your deadline

Late submits

If you got an extension or plan on submitting after the deadline, do NOT submit before the original deadline!

If you accidently did this, it is YOUR responsibility to inform us, Otherwise the latest version before the deadline will be graded!

Questions about this part?

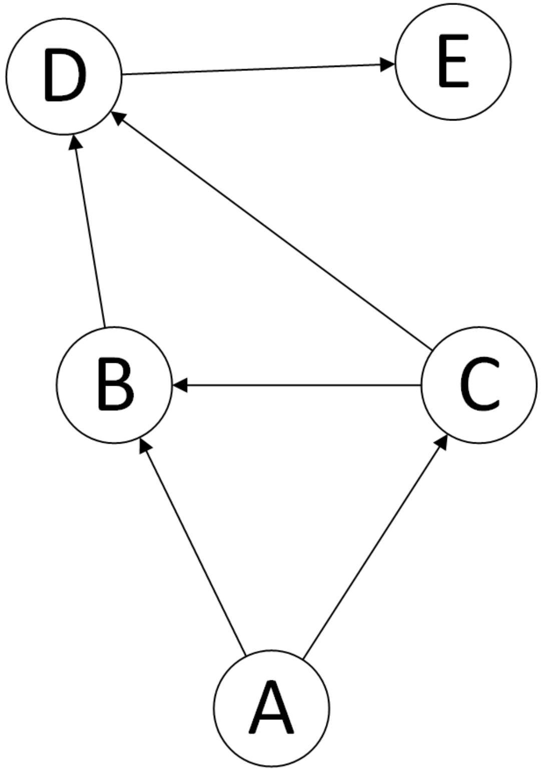


The Assignment

Part 1: Search

Consider: Graph

- Represented as flattened adjacency matrix

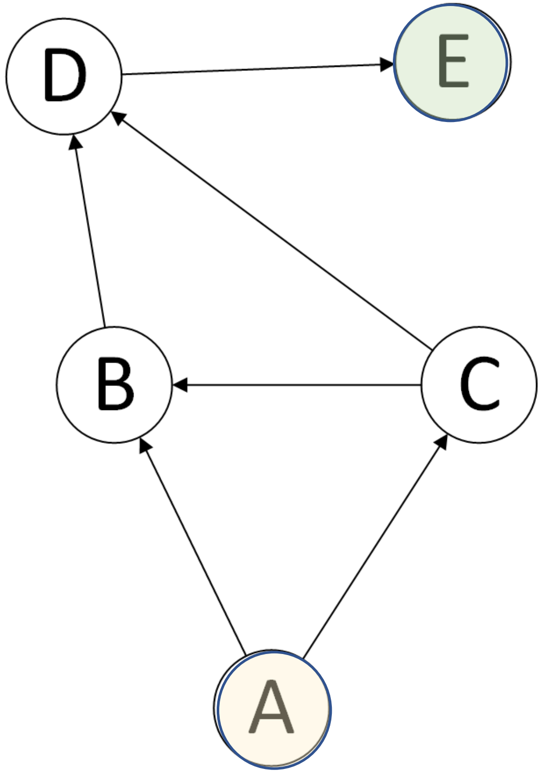


	A	B	C	D	E
A	0	1	1	0	0
B	0	0	0	1	0
C	0	1	0	1	0
D	0	0	0	0	1
E	0	0	0	0	0

Problem: Agent/Search on this Grid

- Goal based Agent
- Wants to go from Start Node to Goal Node
- Can only move along the edges

Your task is to implement
Different search strategies
Such that the agent can reach
the goal if possible.

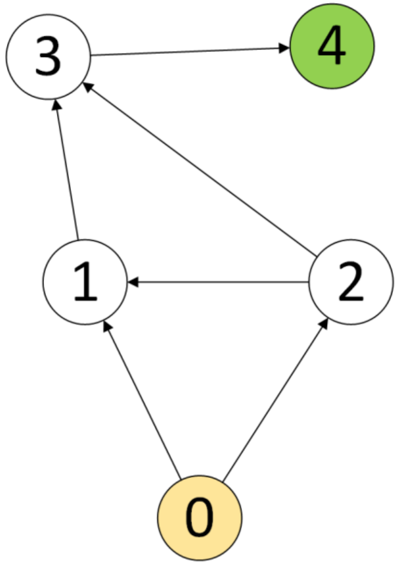


	A	B	C	D	E
A	0	1	1	0	0
B	0	0	0	1	0
C	0	1	0	1	0
D	0	0	0	0	1
E	0	0	0	0	0

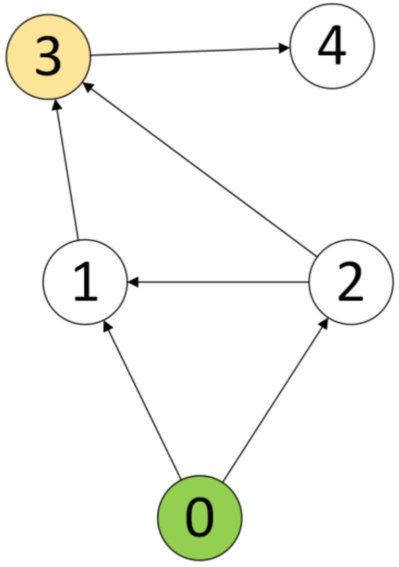
Problem:
Agent searching on Graph

- Random Generation of:
  - Start Node
  - Goal Node
  - Number of Nodes (up to 20)
  - Connections

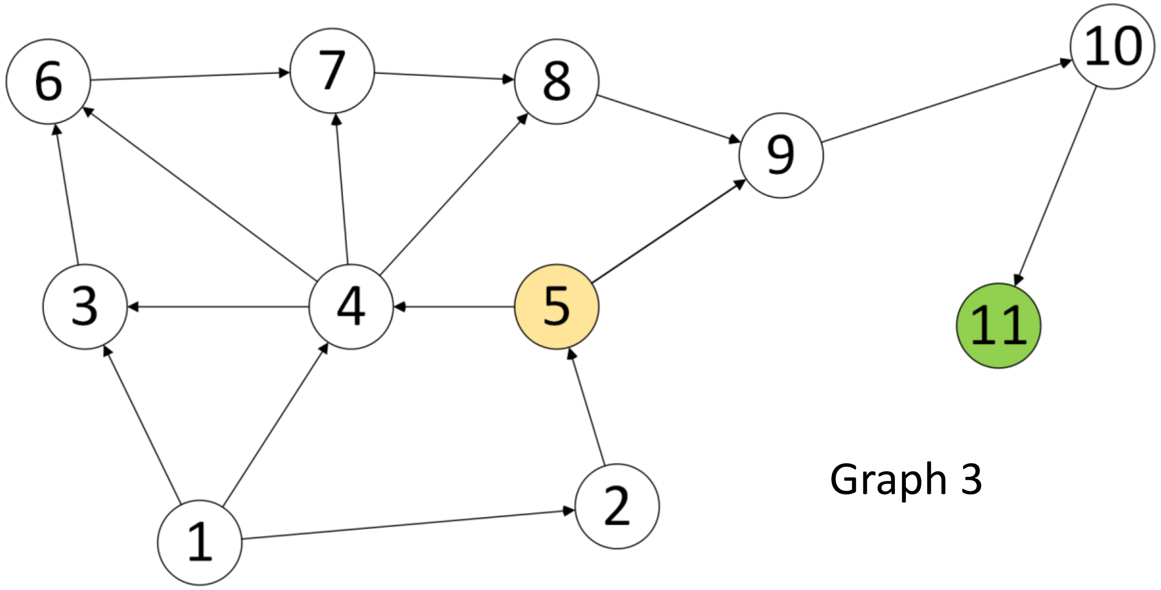
Your task is to implement Different search strategies Such that the agent can reach the goal if it is reachable



Graph 1



Graph 2

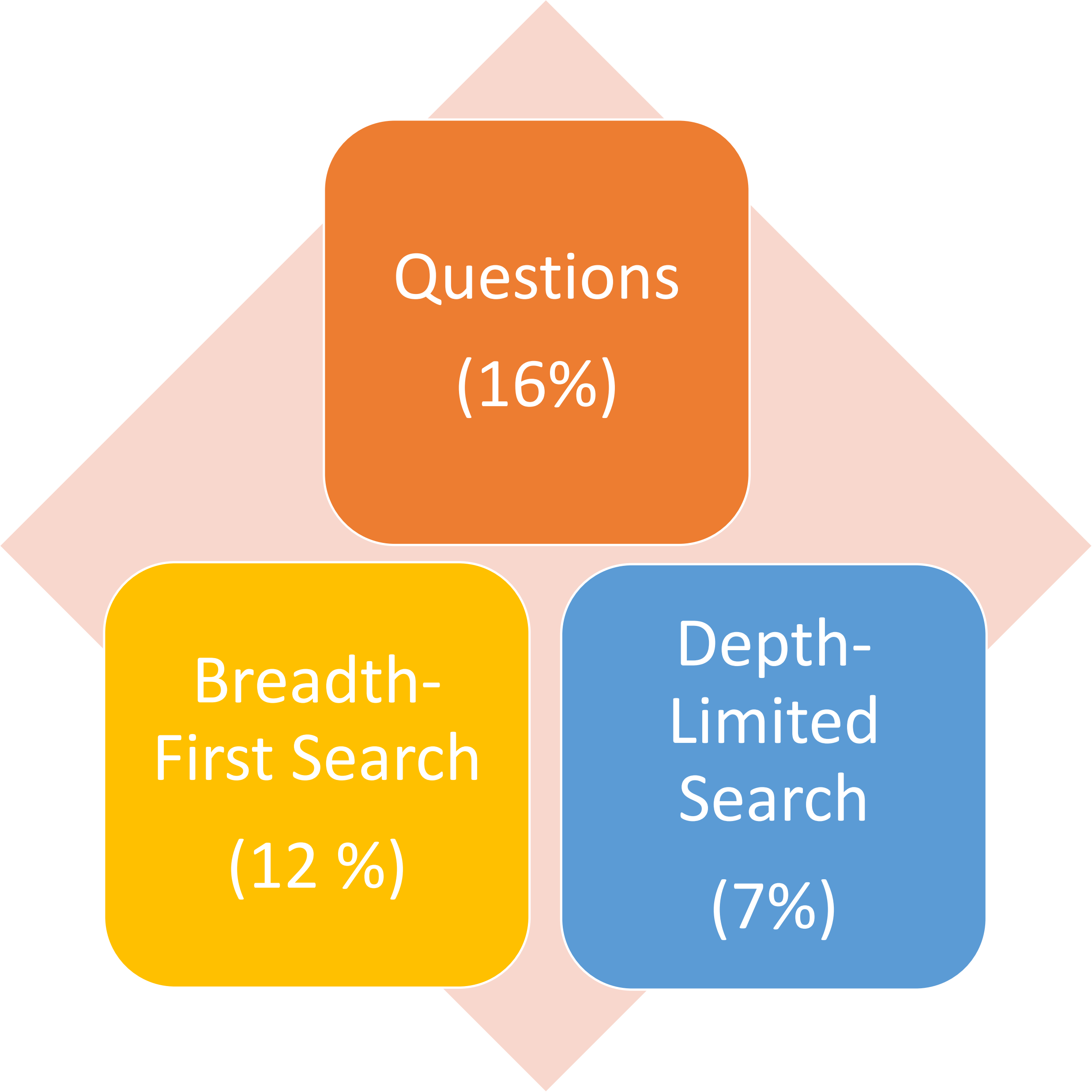


Graph 3

Questions about this problem?

Uninformed Search (35 %)

Search Algorithms



Questions

(16%)

Breadth-First Search

(12 %)

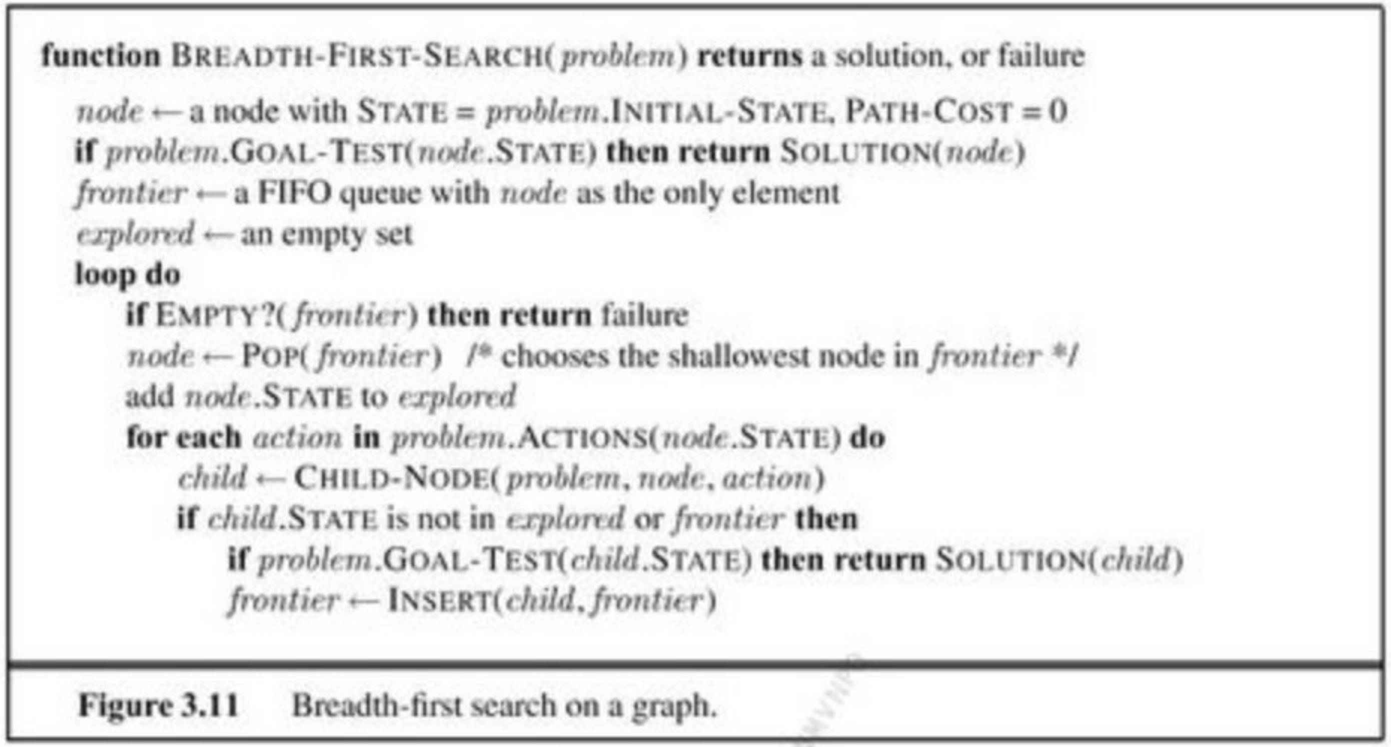
Depth-Limited Search

(7%)

A closer look: Breadth- First Search

Step 1 : Look up and understand the algorithm

We expect you to follow the pseudocode Closely in your implementation!
You might lose points for using more complicated structures.



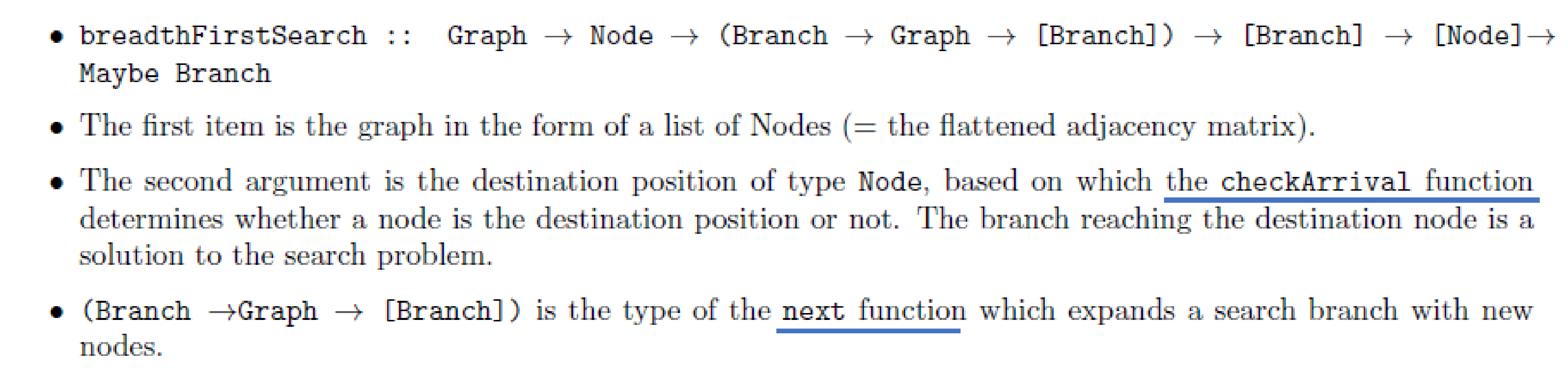
```
function BREADTH-FIRST-SEARCH(problem) returns a solution, or failure
  node ← a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
  if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
  frontier ← a FIFO queue with node as the only element
  explored ← an empty set
  loop do
     if EMPTY?(frontier) then return failure
     node ← POP(frontier)  /* chooses the shallowest node in frontier */
     add node.STATE to explored
     for each action in problem.ACTIONS(node.STATE) do
        child ← CHILD-NODE(problem, node, action)
        if child.STATE is not in explored or frontier then
           if problem.GOAL-TEST(child.STATE) then return SOLUTION(child)
           frontier ← INSERT(child, frontier)
```

Figure 3.11 Breadth-first search on a graph.

A closer look: Breadth- First Search

Step 2 : Carefully read the instructions

Step 3: Start implementing according to them

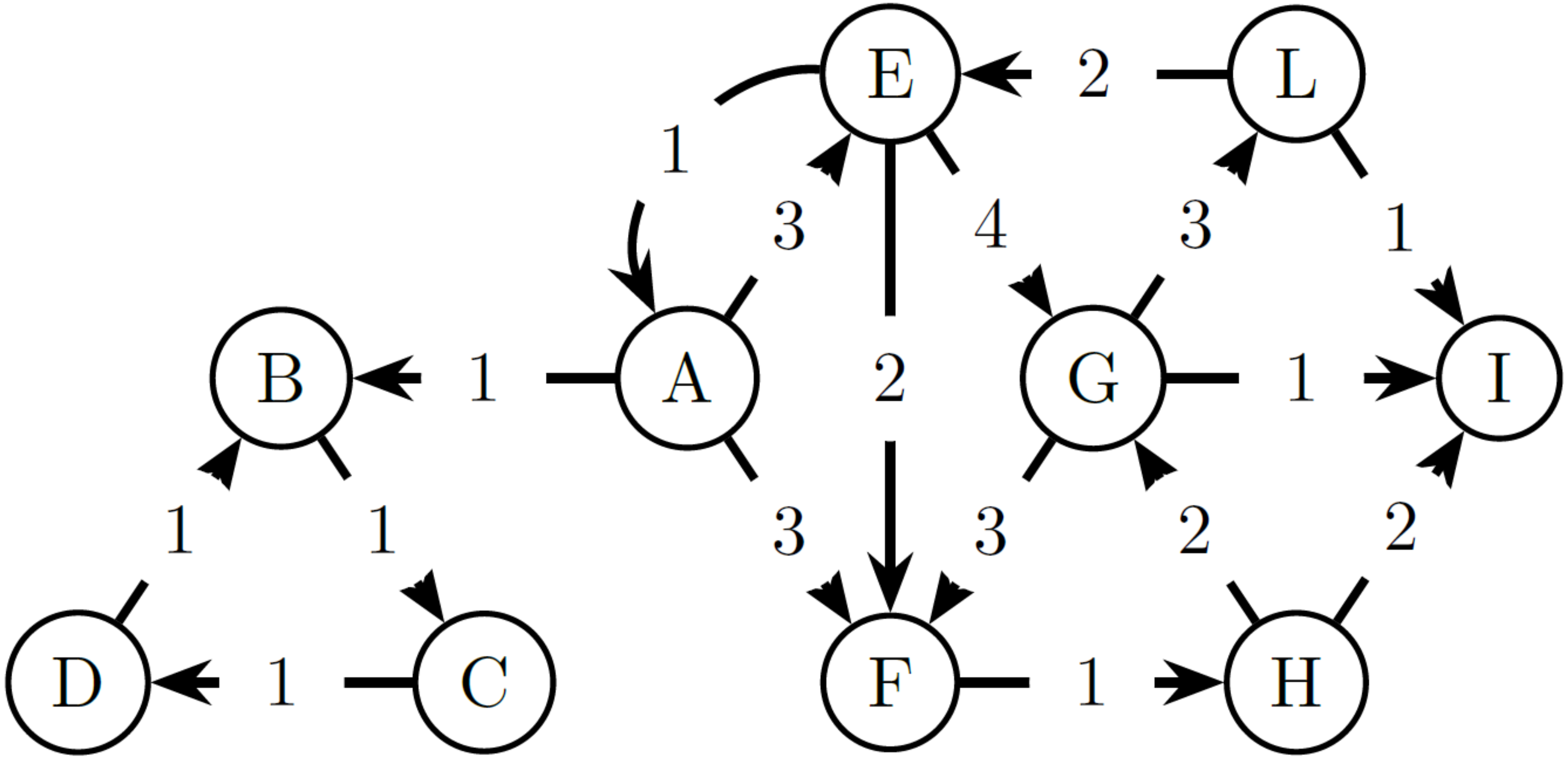


- breadthFirstSearch :: Graph → Node → (Branch → Graph → [Branch]) → [Branch] → [Node]→ Maybe Branch
- The first item is the graph in the form of a list of Nodes (= the flattened adjacency matrix).
- The second argument is the destination position of type Node, based on which the checkArrival function determines whether a node is the destination position or not. The branch reaching the destination node is a solution to the search problem.
- (Branch →Graph → [Branch]) is the type of the next function which expands a search branch with new nodes.

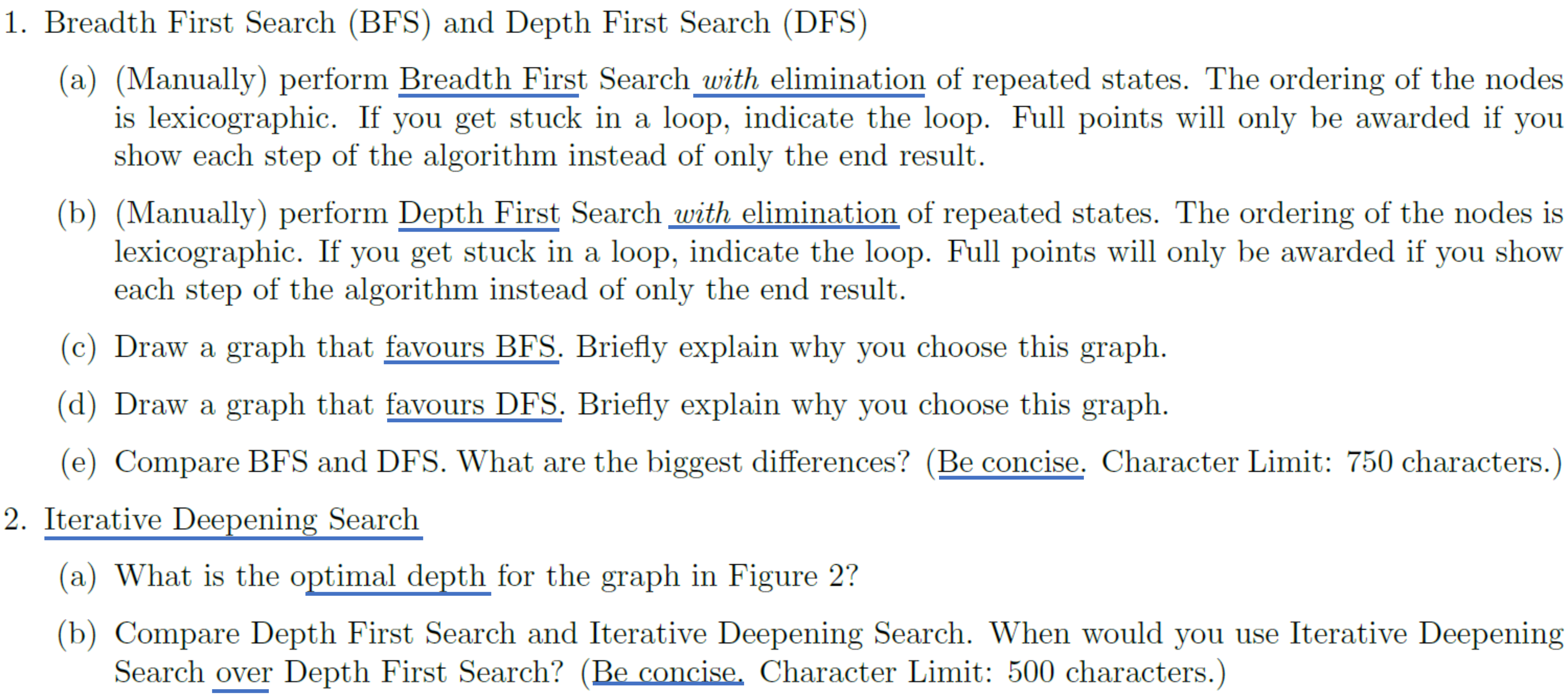
Be on the lookout for hints
In the instructions.

Be careful not to change the
structure of the functions!

Questions: Graph



Questions: Text



1. Breadth First Search (BFS) and Depth First Search (DFS)

   (a) (Manually) perform Breadth First Search *with* elimination of repeated states. The ordering of the nodes is lexicographic. If you get stuck in a loop, indicate the loop. Full points will only be awarded if you show each step of the algorithm instead of only the end result.

   (b) (Manually) perform Depth First Search *with* elimination of repeated states. The ordering of the nodes is lexicographic. If you get stuck in a loop, indicate the loop. Full points will only be awarded if you show each step of the algorithm instead of only the end result.

   (c) Draw a graph that favours BFS. Briefly explain why you choose this graph.

   (d) Draw a graph that favours DFS. Briefly explain why you choose this graph.

   (e) Compare BFS and DFS. What are the biggest differences? (Be concise. Character Limit: 750 characters.)

2. Iterative Deepening Search

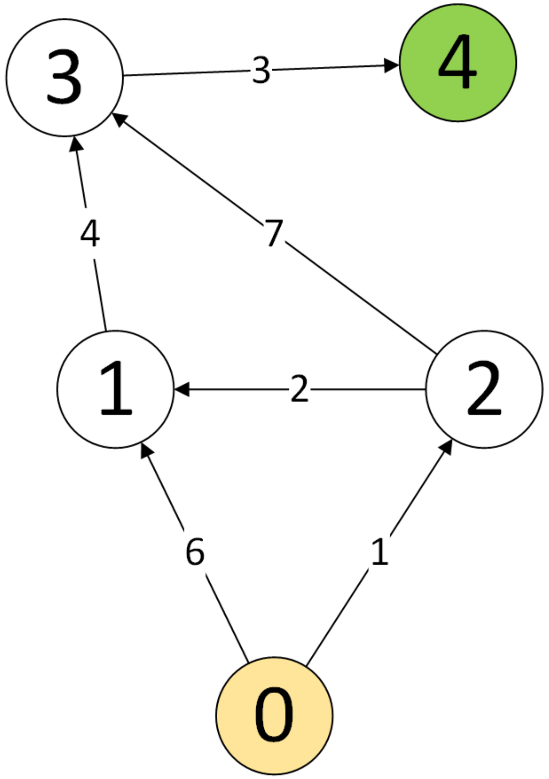
   (a) What is the optimal depth for the graph in Figure 2?

   (b) Compare Depth First Search and Iterative Deepening Search. When would you use Iterative Deepening Search over Depth First Search? (Be concise. Character Limit: 500 characters.)

Informed Search (30%)

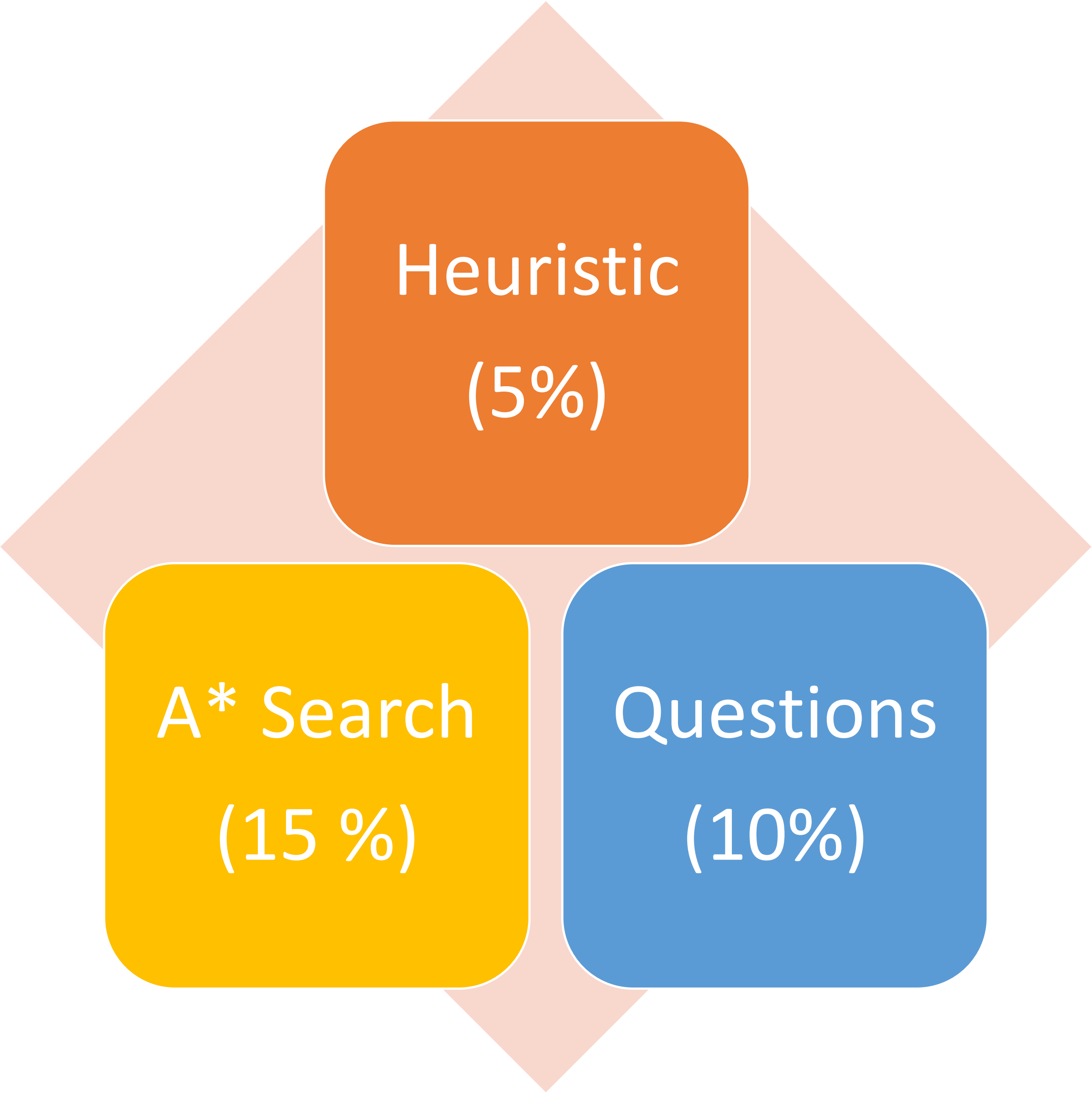
Consider: Graph with cost on edges

- Represented as flattened adjacency matrix



	0	1	2	3	4
0	0	6	1	0	0
1	0	0	0	4	0
2	0	2	0	7	0
3	0	0	0	0	3
4	0	0	0	0	0

Informed Search



Heuristic

(5%)

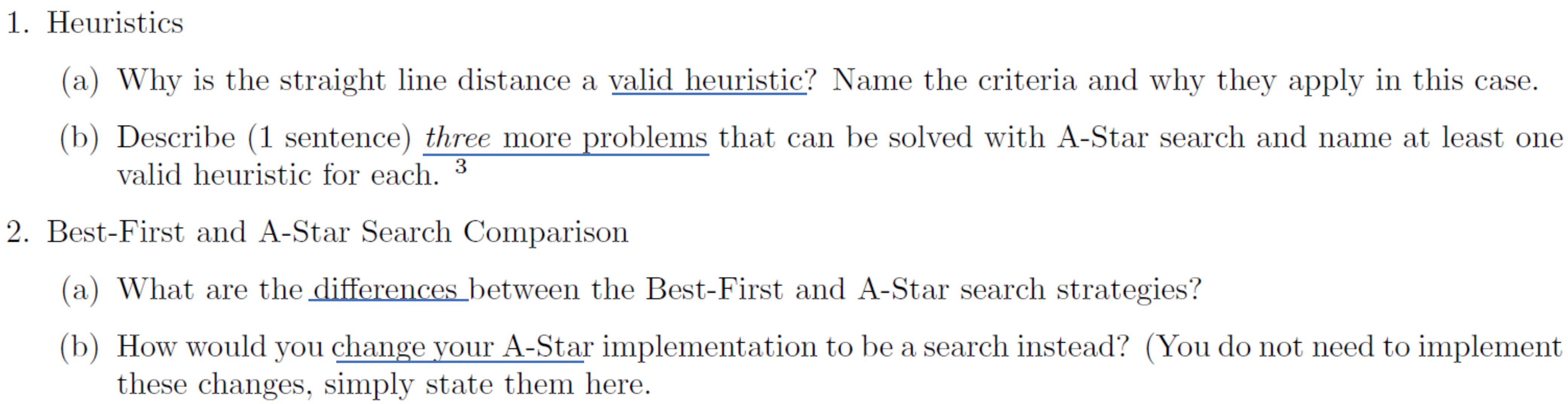
A* Search

(15 %)

Questions

(10%)

Questions



1. Heuristics

   (a) Why is the straight line distance a valid heuristic? Name the criteria and why they apply in this case.

   (b) Describe (1 sentence) *three* more problems that can be solved with A-Star search and name at least one valid heuristic for each. [3]

2. Best-First and A-Star Search Comparison

   (a) What are the differences between the Best-First and A-Star search strategies?

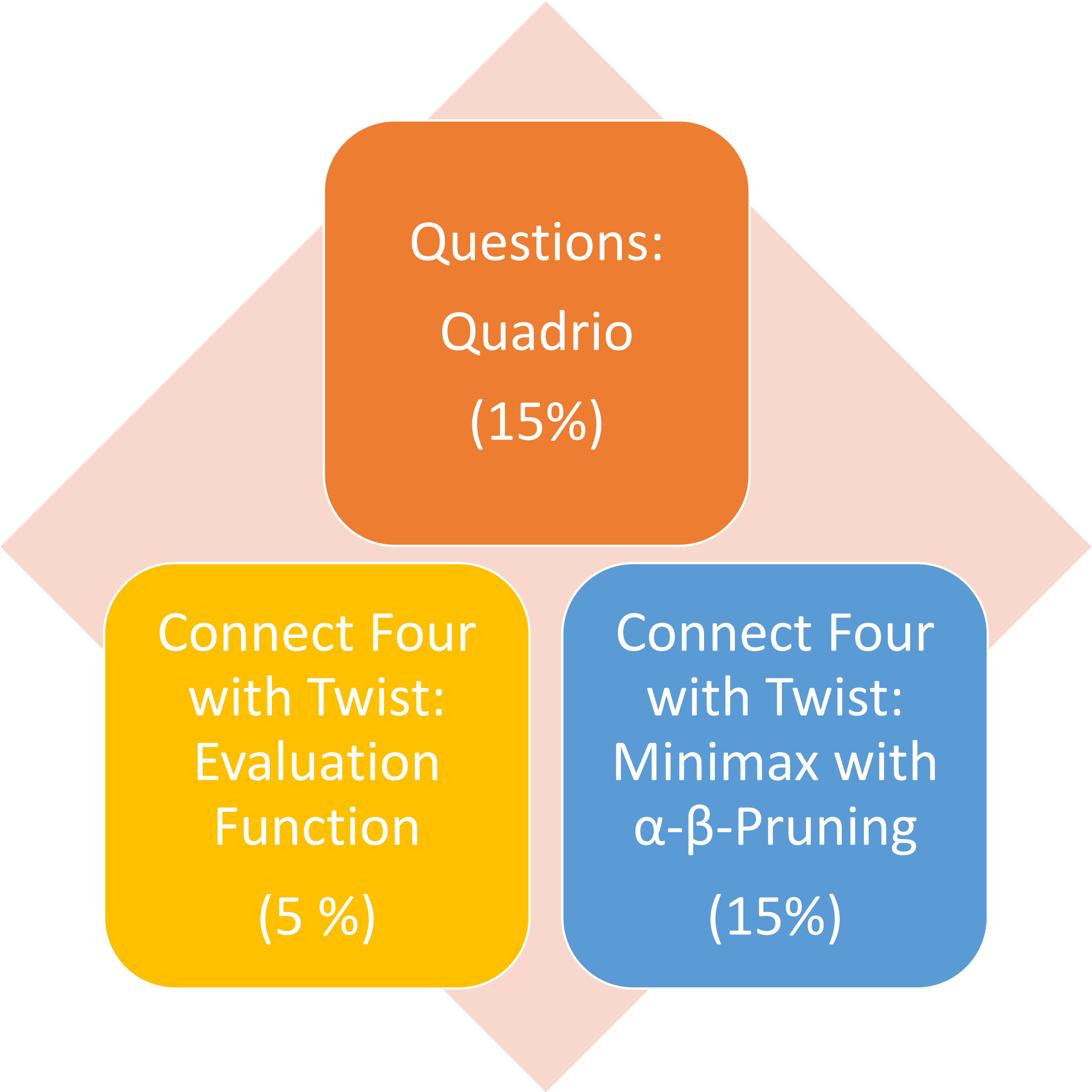
   (b) How would you change your A-Star implementation to be a search instead? (You do not need to implement these changes, simply state them here.

Questions about this part?

Games (35%)

Games:

Task overview



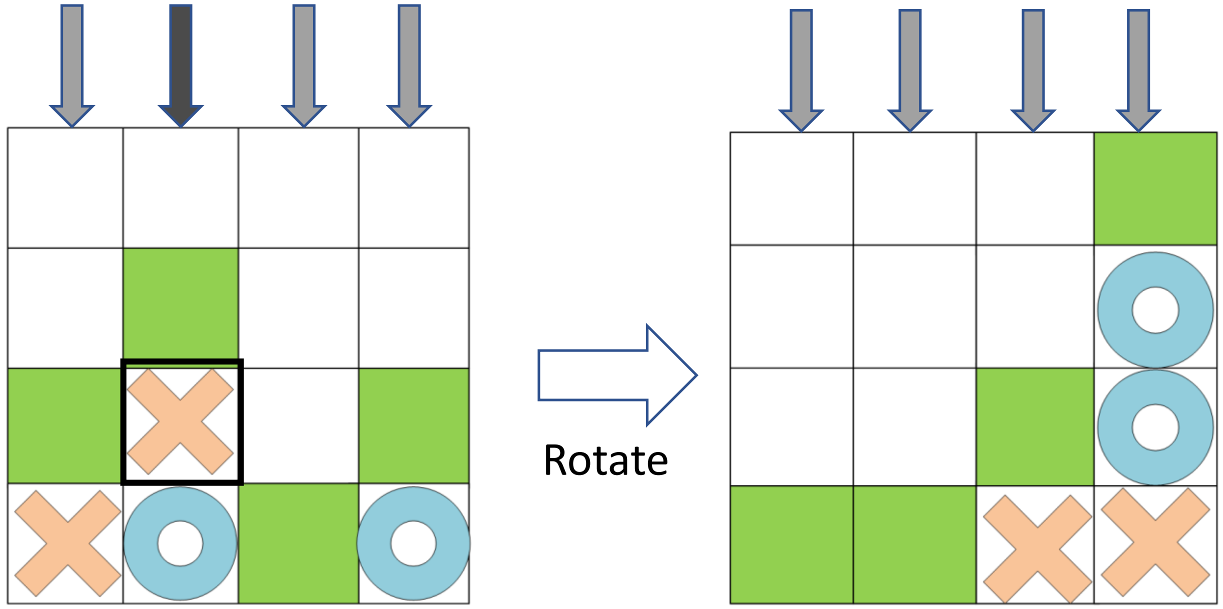
Questions: Quadrio (15%)

Connect Four with Twist: Evaluation Function (5 %)

Connect Four with Twist: Minimax with α-β-Pruning (15%)

Game 1:

Connect Four with a Twist



How to play

At each turn, a player puts their symbol on one of the four slots. The symbol falls down to the next unoccupied square.

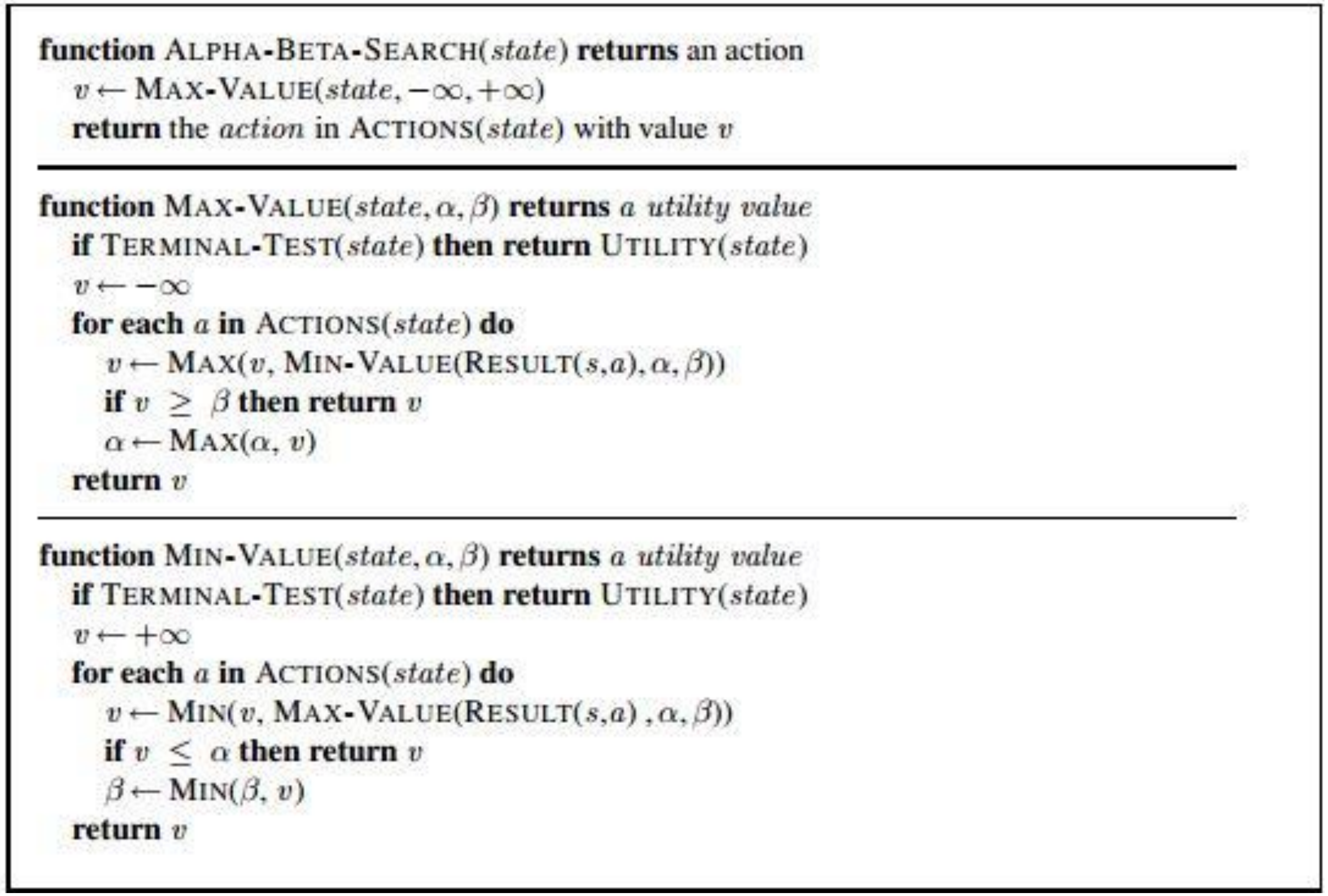
Additionally, the player can choose to rotate the game board to the left. The player does not have to rotate the board.

A player wins if they are able to place 4 of their symbols next to each other vertically, horizontally or diagonally.

Game 1:

Connect Four with a Twist

Minimax with Alpha/Beta Pruning



```
function ALPHA-BETA-SEARCH(state) returns an action
  $v \leftarrow$ MAX-VALUE(state, $-\infty, +\infty$)
  return the action in ACTIONS(state) with value $v$

function MAX-VALUE(state, $\alpha, \beta$) returns a utility value
  if TERMINAL-TEST(state) then return UTILITY(state)
  $v \leftarrow -\infty$
  for each $a$ in ACTIONS(state) do
    $v \leftarrow$ MAX($v$, MIN-VALUE(RESULT($s$,$a$), $\alpha, \beta$))
    if $v \geq \beta$ then return $v$
    $\alpha \leftarrow$ MAX($\alpha$, $v$)
  return $v$

function MIN-VALUE(state, $\alpha, \beta$) returns a utility value
  if TERMINAL-TEST(state) then return UTILITY(state)
  $v \leftarrow +\infty$
  for each $a$ in ACTIONS(state) do
    $v \leftarrow$ MIN($v$, MAX-VALUE(RESULT($s$,$a$) , $\alpha, \beta$))
    if $v \leq \alpha$ then return $v$
    $\beta \leftarrow$ MIN($\beta$, $v$)
  return $v$
```

Don’t worry we have already implemented Connect Four with a Twist and some Helper functions!

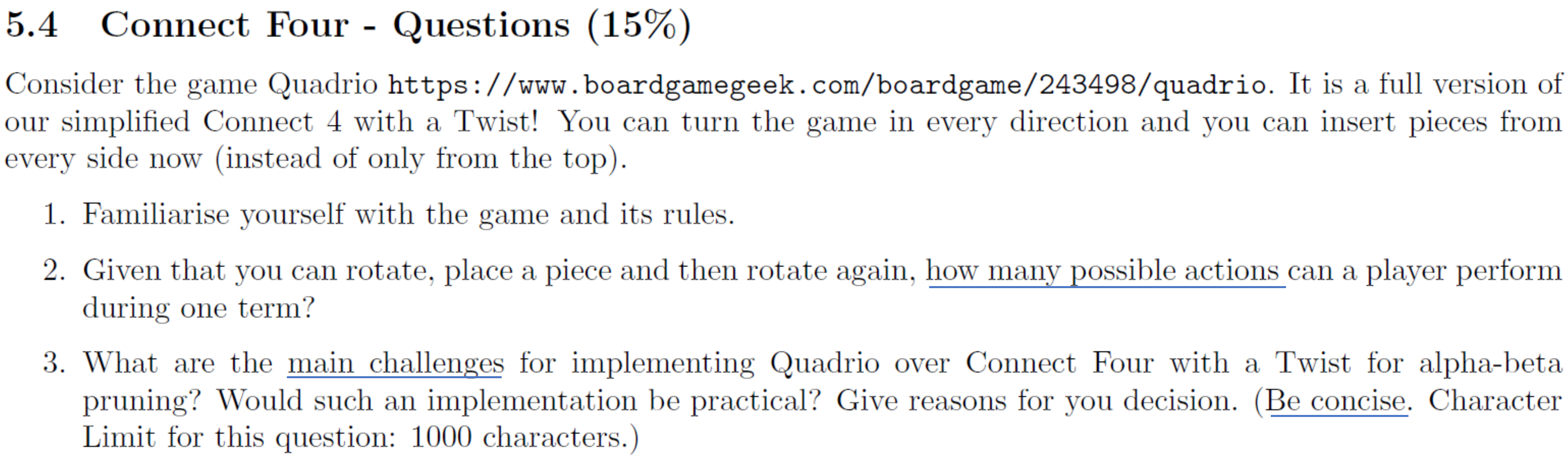
Game 2:

Quadrio

How to play

URL: https://www.youtube.com/watch?v=nY7idZb3GZE&feature=emb_logo

Questions



5.4 Connect Four - Questions (15%)

Consider the game Quadrio `https://www.boardgamegeek.com/boardgame/243498/quadrio`. It is a full version of our simplified Connect 4 with a Twist! You can turn the game in every direction and you can insert pieces from every side now (instead of only from the top).

1. Familiarise yourself with the game and its rules.
2. Given that you can rotate, place a piece and then rotate again, how many possible actions can a player perform during one term?
3. What are the main challenges for implementing Quadrio over Connect Four with a Twist for alpha-beta pruning? Would such an implementation be practical? Give reasons for you decision. (Be concise. Character Limit for this question: 1000 characters.)

That’s it!

Questions?