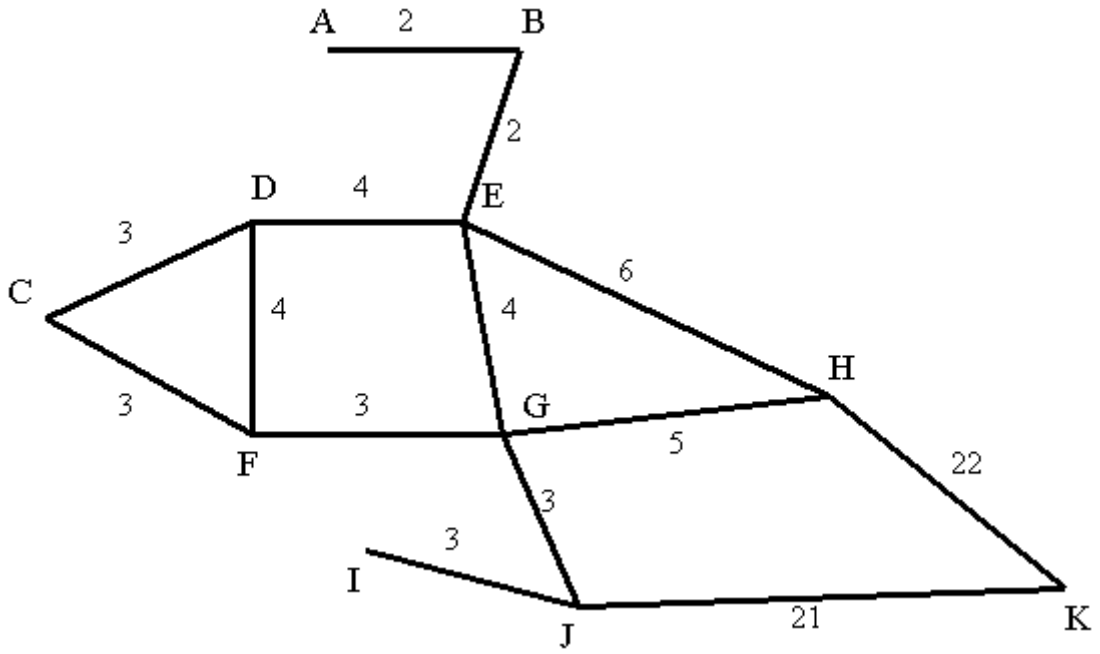
Question 1 Search **(8+6+6+5 = 25 marks)**

Consider the map below (not drawn to scale) with distance between cities shown on the arcs:



Straightline heuristic values for the distance from each city to city **A** are given in the following table:

h(A) = 0	h(B) = 2	h(E) = 3	h(D) = 3	h(C) = 5	h(F) = 6
h(G) = 7	h(H) = 8	h(I) = 8	h(J) = 9	h(K) = 29	

(a) Using the straightline distance heuristic, show the operation of the A* search algorithm in finding the shortest path from city **C** to city **A**. You should construct a heuristic search tree with the nodes annotated on the side, showing the following:

1. the order of expansion of the nodes (e.g., a circled number 3 marked beside a node indicates the node is the 3rd to be expanded). **(8 marks)**

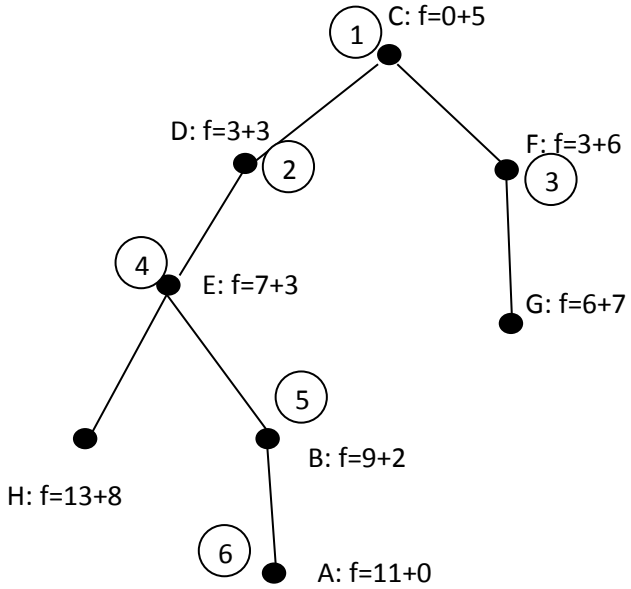
2. at each step of the expansion, the nodes in the OPEN and CLOSED lists, until a solution is found. **(6 marks)**

3. the evaluation function *f*-value of each node in the search graph. **(6 marks)**

(b) Would you expect the A* search using the straightline distance heuristic to be better than a Breadth-First Search (BFS)? Explain why briefly. **(5 marks)**

Sample solution:

a.



The open and closed sets at each step of node expansion:

1. Open=[C5]; Closed=[ ]
2. Open=[D6,F9]; Closed=[C5]
3. Open=[F9,E10]; Closed=[C5,D6]
4. Open=[E10,G13]; Closed=[C5,D6,F9]
5. Open =[B11,G13,H21]; Closed=[C5,D6,F9,E10]
6. Open=[A11,G13,H21]; Closed=[C5,D6,F9,E10,B11]
7. Goal found!

The order of node expansion therefore is: **CDFEBA**.

b. Yes, because A* using straightline distance heuristic will expand fewer nodes than a breadth-first search does, therefore it is more efficient. A* will always find the shortest path to the goal, as it is complete (ie., always finds a solution) and optimal (always find the best solution - the shortest path to goal).

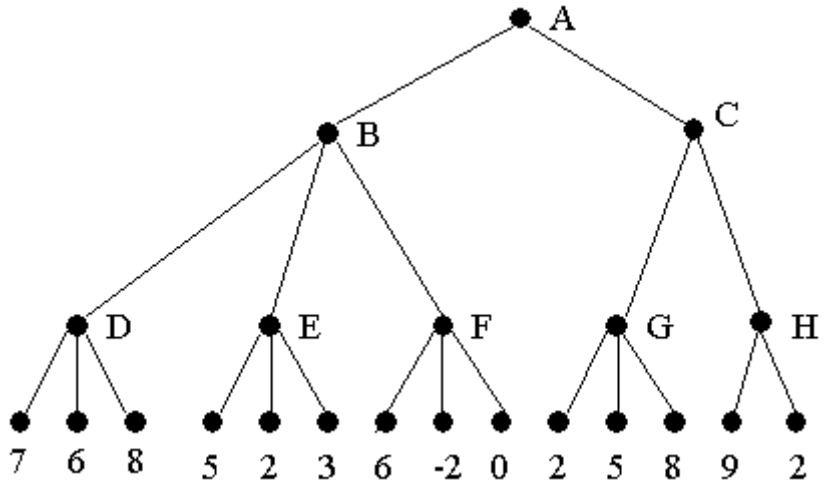
Question 2 Search and game playing (3+3+4+5 = 15 marks)

(a) Define the concept of "admissibility" of a heuristic. Describe how admissibility relates to the A* algorithm. **(3 marks)**.

(b) Describe a search space in which Iterative Deepening Search (ITS) performs much worse than Depth-First Search (DFS). **(3 marks)**

(c) Assuming MAX is about to move in the game tree below, which move will be chosen if minimax search is used? **(4 marks)**

(d) Redraw the game tree below showing ONLY the nodes expanded by an alpha-beta search and showing clearly the ORDER of evaluation of the nodes. **(5 marks)**.

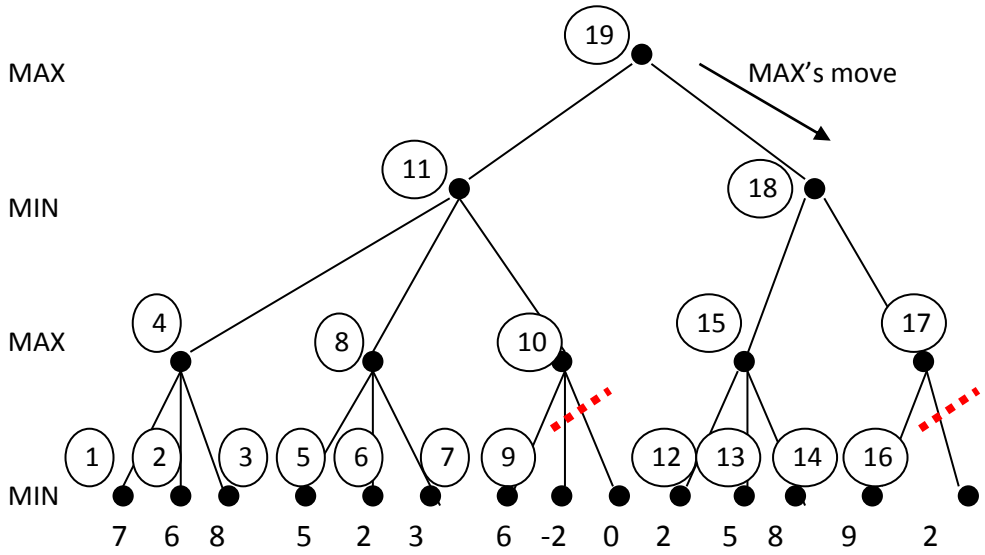


Sample solutions:

a) Admissibility of a heuristic says that the heuristic will never overestimate the cost to reach the goal. For example, straightline distance heuristic is admissible, because between any two points A and B, the shortest distance is a straightline, which is always equal or less than the actual distance. Other example of admissible heuristics include the number of titles out-of-place and city block heuristics we use for the 8-puzzle problem. A* search uses $f(n) = g(n) + h(n)$, which combines a measurement of the distance from the initial state to the current state n, $g(n)$, with an admissible heuristic measuring the distance from the current state n to the goal state, $h(n)$. By using an admissible heuristic for h, A* is complete and optimal, which means it can always find the shortest path from the initial to the goal state.

b) Iterative deepening search (IDS) combines the benefit of both depth-first and breadth-first search by trying all possible depth limits, first to depth 0, then depth 1, 2, and so on. It expanded many states multiple times. For a search space with a branching factor of 1, for example the figure below, an IDS will expand many more times repetitively the states it has already expanded previously, whereas a depth-first search will only take n expansions to find a solution at depth n.

c) Max will choose to move towards node C. See the figure below.



d) The order of evaluations is shown in the above figure.

Question 3 Logic (3+5+3+4 = 15 marks)

(a) Using truth tables, show that $P \Rightarrow Q$ is logically equivalent to $\neg P \vee Q$. **(3 marks)**

(b) Consider the following story:

"All people that are not poor and are smart are happy. Those people that read are smart. John is wealthy. Helen can read and is wealthy. Happy people have exciting lives. Wealthy people are not poor."

1) Translate the story into first order logic expressions. **(5 marks)**

2) Determine who has an exciting life using rules of inference in first order logic. **(3 marks)**

3) Show the solution process with the AND/OR graph. **(4 marks)**

Sample solutions:

(a)

P	Q	¬P	P⇒Q	¬P ⋁ Q
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

The last two columns on the right have the same truth values, therefore the above two logical expressions are equivelant.

(b)

1.

R1: ∀X (¬poor(X) /\ smart(X) ⇒ happy(X))

R2: ∀X (read(X) ⇒ smart(X))

wealthy(john)

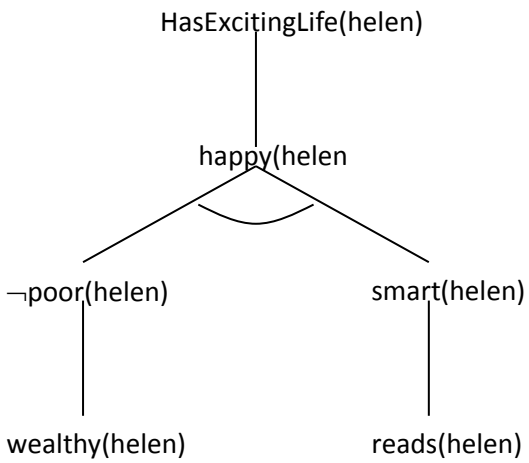
reads(helen)

wealthy(helen)

R3: ∀X (happy(X) ⇒ hasExcitingLife(X)

R4: ∀X (wealthy(X) ⇒ ¬poor(X))

The AND/OR graph for the problem solution is:



2. We can use either forward or backward chaining to do this. If using forward chaining, starting with the known facts:

reads(helen)
wealthy(helen)

then by R2, we know:

smart(helen)

and by R4, we know:

¬poor(helen)

and then with the above two new facts, by R1, we get:

happy(helen)

and finally we can apply R3 to assert that:

hasExcitingLife(helen)

However HasExcitingLife(john) would fail, because we cannot prove reads(john), hence smart(john). We cannot therefore prove happy(john) to be TRUE.

3. See the AND/OR graph above. We start from the top, trying to prove the goal, and this requires sub-goals to be asserted. This process continues until we reach to the leaf nodes of the tree, which are linked with the facts in our knowledge base. Once the truth values of these facts are found, we can work our way back and prove the goal to be true or not. As the graph shows, we can prove helen has an exciting life by following this backward chaining process.

Question 4 Planning (6+3 +4 +2 = 15 marks)

A kitchen cleaning robot plans how to clean a kitchen. There is a fridge in the kitchen. The kitchen is considered cleaned if the fridge and the floor are all cleaned, and the kitchen has no garbage. The robot is smart enough that it only cleans or washes things when they are dirty, and only takes out garbage when there is garbage. The constraints for cleaning the kitchen are as follows:

- Cleaning the fridge will get the floor dirty.
- Cleaning the fridge generates garbage.
- Before the floor can be wasted, it must be swept.
- Before the floor can be swept, the garbage must be taken out.

There are 4 actions that the kitchen cleaning robot can take:

- *clean(fridge)*
- *wash(floor)*
- *sweep(floor)*
- *takeout(garbage)*

The vocabulary of predicates you should use is:

- *clean(X)* – where X is cleaned. X can be either fridge or floor.
- *dirty(X)* – where X is dirty. X can be either fridge or floor.
- *swept(floor)* – where floor has been swept.
- *hasGarbage(kitchen)* – indicates whether the kitchen has garbage.

The corresponding STRIPS definition of the *wash(floor)* action is:

	P: *dirty(floor)* $\wedge$ *swept(floor)*
wash(floor)	A: *cleaned(floor)*
	D: *dirty(floor)* $\wedge$ *swept(floor)*

(a) Write the STRIPS operators for the other three actions, **using only the predicates given above**. **(6 marks)**
(b) The goal for the robot is to clean the kitchen, and the initial state of the kitchen is: the fridge is dirty, the floor is cleaned, and there is no garbage in the kitchen. Using only the predicates given above, write down descriptions of the initial state and the goal state. **(3 marks)**
(c) Show how a plan can be generated to solve the problem described in (b). **(4 marks)**
(d) Given your plan, in what order will the actions be executed by the robot? **(2 marks)**

Sample solutions:

a.

	P: *dirty(fridge)*
clean(fridge)	A: *cleaned(fridge) /\ hasGarbage(kitchen) /\ dirty(floor)*
	D: *dirty(fridge) /\ cleaned(floor) /\ hasGarbage()*

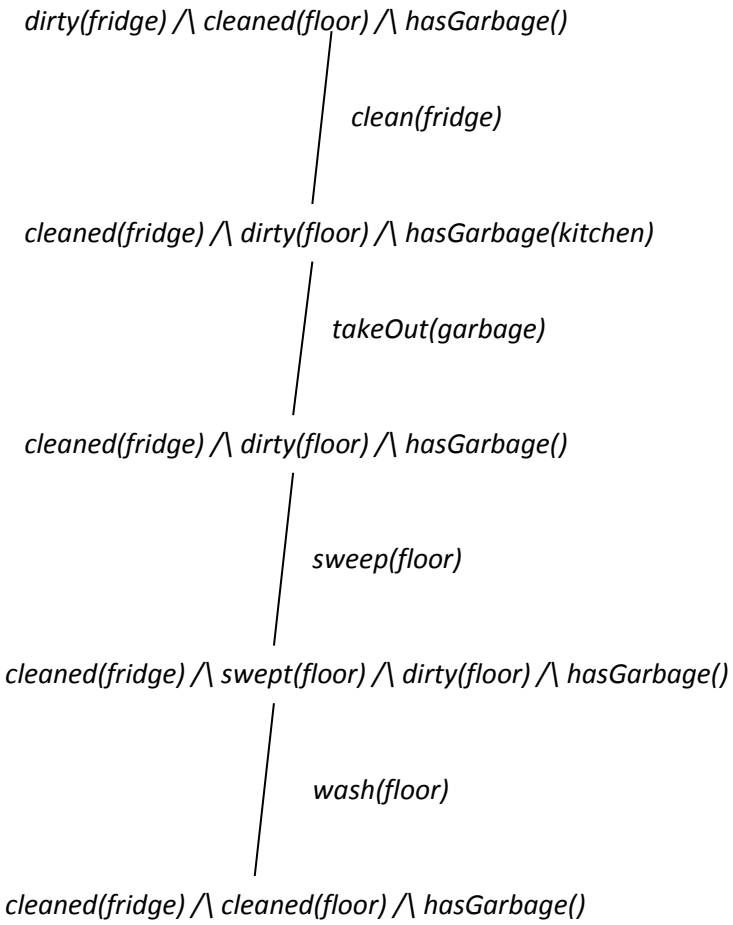
P: *hasGarbage()*
sweep(floor) A: *swept(floor)*
D:

P: *hasGarbage(kitchen)*
takeOut(garbage) A: *hasGarbage()*
D: *hasGarbage(kitchen)*

* **Note**: in the above, if negation is used, we consider it is ok for this exam, although generally speaking, STRIPS would not use negation as it simply adds or deletes predicates that are already there in a state.

b. Initial state: dirty(fridge) /\ cleaned(floor) /\ hasGarbage()
Goal state: clean(fridge) /\ cleaned(floor) /\ hasGarbage()

c.



*Note: again it is ok if negation used in the above, instead of *hasGarbage()*.

Because of the contraints imposed, the above is the complete search space which is very small. Simple blind search such as BFS or DFS will find the path to the goal state easily. All actions taken along this path will constitue a plan to achieve the goal.

d.

The plan is: clean(fridge), takeOut(garbage), sweep(floor) and wash(floor).

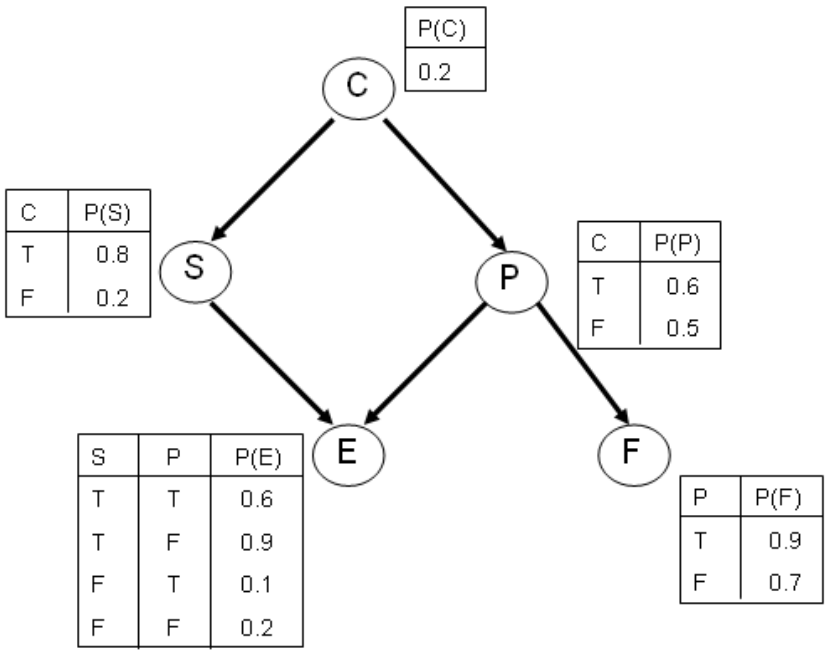
Question 5 Probability and Bayesian Networks (5+5+6+6 = 22 marks)

(a) David is a professional poker player. At the moment, he wants very much to draw two diamonds in a row. As he sits at the table looking at his hand and at the upturned cards on the table, David sees 11 cards. Of these, 4 are diamonds. The full deck contains 13 diamonds among its 52 cards, so 9 of the 41 unseen cards are diamonds. Because the deck was carefully shuffled, each card that David draws is equally likely to be any of the cards that he has not seen. What is David's probability of drawing two diamonds? Show your workings. **(5 marks)**

(b) A laboratory blood test is 99% effective in detecting a certain disease when it is, in fact, present. However, the test also yields a "false positive" result for 1% of the healthy persons tested (that is, if a healthy person is tested, then, with probability 0.01, the test result will imply he has the disease). If 0.5% of the population actually has the disease, what is the probability a person has the disease given that his test result is positive? Show your workings. **(5 marks)**

(c) The following figure shows a simple Bayesian Network, with each node representing the following statements:

C = that you will go to college
S = that you will study
P = that you will party
E = that you will be successful in your exams
F = that you will have fun



Given the CPT for each node, complete the following:

(d) Calculate the probability that you will go to college and that you will study and be successful in your exams, but will not party or have fun. Show your workings. **(6 marks)**

(e) What is the probability that you will have success in your exams if you have fun and study at college, but don't party? Show your workings. **(6 marks)**

Sample solutions:

(a)

P(first card diamond) = 9 / 41

P(second card diamond | first card diamond) = 8 / 40

Using the product rule, we have:

P(both cards diamonds) = P(second card diamond | first card diamond) x P(first card diamond)

= 9/41 x 8/40

= 0.044

(b)

P(disease | positive) = P(disease $\wedge$ positive) / P(positive)

= P(positive | disease) P(disease) / (P(positive | disease)P(disease) + P(positive | $\neg$disease)P($\neg$disease))

= (0.99 x 0.005) / ((0.99 x 0.005) + (0.01 x 0.995))

= 0.3322

(c)

i.

P(C,S,$\neg$P,E, $\neg$F) = P(C) x P(S|C) x P($\neg$P|C) x P(E|S,$\neg$P) x P($\neg$F|$\neg$P)

= 0.2 x 0.8 x 0.4 x 0.9 x 0.3

= 0.01728

ii.

Since E is independent from F and C (given that we know S and P), we have the following:

P(E|F,$\neg$P,S,C) = P(E|S,$\neg$P)

= 0.9

6. Genetic algorithms and machine learning (4+4 = 8 marks)

(a) Write the pseudocode for a standard genetic algorithm. **(4 marks)**
(b) Identify and briefly discuss at least three similarities and differences between genetic algorithms and genetic programming. **(4 marks)**

Sample solutions:

a.

Procedure Genetic Algorithm:

```
begin:
    set time t := 0;
    initialize the population P(t);
    while the termination condition is not met do
        begin
            evaluate fitness of each individual in the population P(t);
            select individuals from population P(t) based on fitness;
            produce the offspring of these pairs using genetic operators such as crossover and mutations;
            replace, based on fitness, candidates of P(t), with these offsprings;
            set time t := t + 1;
        end
end.
```

b.

Both GAs and GP are based on the principle 'survival of the fittest', of Dawin's evolutionary theory. They both reply on evolving a population of potential solutions. It normally starts with an initiall randomly generated individuals, and then assign fitness to each individual according to some criteria. This happen over many generations, and at each generation, genetic operators such as crossover and mutations are applied to generate new offspring. Fitter individuals are favored over those less fit ones each time, therefore over successive generations, the population will get fitter until a stop criteria is met, and then we can take the fittest individual in the population as the best solution to the problem being optimized.

However GP is different from GA in the following aspects:

- GP starts with an initial population of randomly generated programs, instead of just binary strings often used in a GA population.

- These programs often can be represented as tree structures, made up of appropriate program pieces such as arithmetic operators, simple math functions, as well as logical and domain specific functions. Programs may also contain components including data items of the usual data types: boolean, integer, floating point, vector, symbolic, or multi-valued.

- Crossover is carried out often by swapping certain branches of two parents, so that two new programs (offspring) can be generated. A GP system often check the validity of the newly generated programs to make sure that they are valid syntactically. Mutation is simply to replace a branch of a tree with another randomly generated branch. In GAs, we often apply crossover and mutations on fixed binary strings

- The fitness of a program is determined by how well it performs a certain task, e.g., to measure the output of the program and see how well it approximates the targeted output. In Gas, it is more general, depending on the fitness function identified.

End of exam paper