2(b)
Suppose we have the following dataset describing three plant species:

	Species 1				
Altitude (100m)	2.20	2.04	1.95	1.79	1.91
Height (cm)	1.99	2.19	2.23	2.22	2.58

	Species 2				
Altitude (100m)	2.31	2.49	2.94	2.20	1.97
Height (cm)	2.53	2.20	1.88	2.68	2.93

	Species 3				
Altitude (100m)	1.02	1.54	0.82	2.93	0.76
Height (cm)	1.15	0.95	1.05	0.97	1.10

(i) Build a Gaussian naive Bayes classifier using this dataset and classify the point (2.1, 2.3). Show your working.

First of all, we need to calculate the mean and standard deviation for each feature in all classes. The following table shows the results:

Feature	Altitude		Height	
Class	Mean	SD	Mean	SD
Species 1	1.97	0.14	2.24	0.19
Species 2	2.38	0.33	2.44	0.37
Species 3	1.41	0.81	1.04	0.08

$\vec{x} = (2.1, 2.3)$

Species 1

P(C1) = $\frac{5}{30}$ = 0.166

P(X altitude | C1) = $\frac{1}{0.14\sqrt{2\pi}}$ exp (- $\frac{1}{2}$ ($\frac{2.1-1.97}{0.14}$)^2)

P(X height | C1) = $\frac{1}{0.19\sqrt{2\pi}}$ exp (- $\frac{1}{2}$ ($\frac{2.3-2.24}{0.19}$)^2)

Log(P(Ci)) + $\sum_{j=1}^{n} \mathrm{Log}(\mathrm{P}(\mathrm{xj}\ |\mathrm{ci}))$ = Log(P(C1)) + Log P(X altitude | C1) + Log P(X height C1)

= Log(0.166) + Log(0.63669) + Log(0.96368)

= —0.78 — 0.20 — 0.02

= —1

Species 2

P(C2) = 0.166

P(X altitude | C2) = $\frac{1}{0.33\sqrt{2\pi}}$ exp (- $\frac{1}{2}$ ($\frac{2.1-2.38}{0.33}$)^2)

P(X height | C2) = $\frac{1}{0.37\sqrt{2\pi}}$ exp (- $\frac{1}{2}$ ($\frac{2.3-2.44}{0.37}$)^2)

Log(P(Ci)) + $\sum_{j=1}^{n}$ Log(P(xj |ci)) = Log(P(C2)) + Log P(X altitude | C2) + Log P(X height C2)

= Log(0.166) + Log(0.93398) + Log(0.99462)

= —0.78 — 0.03 — 0.002

= —0.812

Species 3

P(C3) = 0.166

P(X altitude | C3) = $\frac{1}{0.81\sqrt{2\pi}}$ exp (- $\frac{1}{2}$ ($\frac{2.1-1.41}{0.81}$)^2)

P(X height | C3) = $\frac{1}{0.08\sqrt{2\pi}}$ exp (- $\frac{1}{2}$ ($\frac{2.3-1.04}{0.08}$)^2)

Log(P(Ci)) + $\sum_{j=1}^{n}$ Log(P(xj |ci)) = Log(P(C3)) + Log P(X altitude | C3) + Log P(X height C3)

= Log(0.166) + Log(1.29299) + Log(2.80869)

= —0.78 + 0.11 + 0.45

= —0.22

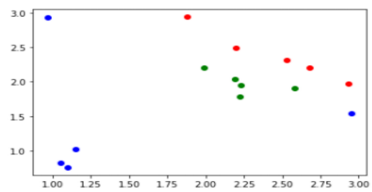
Since Class 3 (Species 3) maximises the quantity log(P (ci)) + $\sum_{j=1}^{n}$ log(P (xj |ci)), so we classify $\vec{x}$ = (2.1, 2.3) as belonging to class 3 (Species 3).

(ii) Do any of the classes fit the independence assumption? Do any of the classes violate the independence assumption? Explain your answer. (Hint: you may wish to plot the data)

The following scatter plot shows which classes fit the independence assumption:

```
import matplotlib.pyplot as plt
plt.scatter(df['hight'][:5],df['altitude'][:5],color = 'g')
plt.scatter(df['hight'][5:10],df['altitude'][5:10],color = 'r')
plt.scatter(df['hight'][10:15],df['altitude'][10:15],color = 'b')
plt.show
```

<function matplotlib.pyplot.show(close=None, block=None)>



- The first class (species 1) with the green color fit the independence assumption. All points are grouped.
- The third class (species 3) with the blue color violates the independence assumption. The points are not grouped.

2(c)

Would k-means do a good job at clustering this data? Explain why or why not.

We cannot use k-means with this data because k-means for clustering works with unlabelled data. And in this case, we notice that we have a labelled data.

(iv) Uniform cost search

H, C, A, B, Q, G

Explored	Frontier	Paths
H	A, B, C	HC (1.7), HA (1.8), HB (1.8)
HC	U, A, B	HA (1.8), HB (1.8), HCU (3.2)
HCA	Q, U, B	HB (1.8), HAQ (2.0), HCU (3.2)
HCAB	Q, U	HAQ (2.0), HBA (2.8), HCU (3.2)
HCABQ	U, G	HBA (2.8) – A has expanded, HAQG (2.9), HCU (3.2)
HCABQG	U	HCUG (2.2), HAQG (2.9)

Which, if any, of these techniques gives you the optimal route to the museum?

The best search technique that gives the optimal route to the museum is Uniform cost search.

4(b)

Is the heuristic:

1. Admissible?
2. Consistent?

Explain your answer.

Heuristic is admissible, the search can be done according to the length of the path, choosing the shortest path, and knowing that heuristics are methods that give results close to the minimum or the best results for that they are acceptable.

4(c)

Suppose the edge between the student union (node U) and the city museum (node G) now has a negative cost of -1 (because you particularly want to see the fountain on the way). Suppose you try to run uniform cost search on this graph (even though one edge is negative). Does uniform cost search still find the optimal path? If not, why not?

NB: do not simply answer that uniform cost search requires positive edge costs - please investigate what happens in this situation.

In this scenario the uniform cost search will not find the optimal path since the goal stage will be achieved with the path (H, A, Q, G) with a value of 2.9 when node Q is explored. At this point, the value of other explored path (H, C, U) will have a value of 3.2 and the unform cost search will not expand node U.

if we expand U then the path (H, C, U, G) will have a value of 2.2 which will be the optimal path in this case.

4(d)

Now suppose the edge between U and G has a cost of 0.7 as before. You discover that there is a ferry between SS Great Britain (B) and the Aquarium (Q), adding an edge between B and Q at negative cost of -2 (your friend loves ferries). Suppose you try to run uniform cost search on this graph (even though one edge is negative). Does uniform cost search still find the optimal path? If not, why not?

NB: do not simply answer that uniform cost search requires positive edge costs - please investigate what happens in this situation.

If we add an edge between B and Q the new path (H, B, Q, G) will be the shortest path with a value of 0 in distance and it is not applicable. The algorithm will give us the path HBQG which is not applicable.

Question 5:

Consider the 8-puzzle problem given the initial state shown in figure 2. Assume that the space moves up if possible, left if up is not possible, down if neither up nor left are possible, otherwise right (i.e. the order of priority of the moves is up, left, down, right).

5(a)

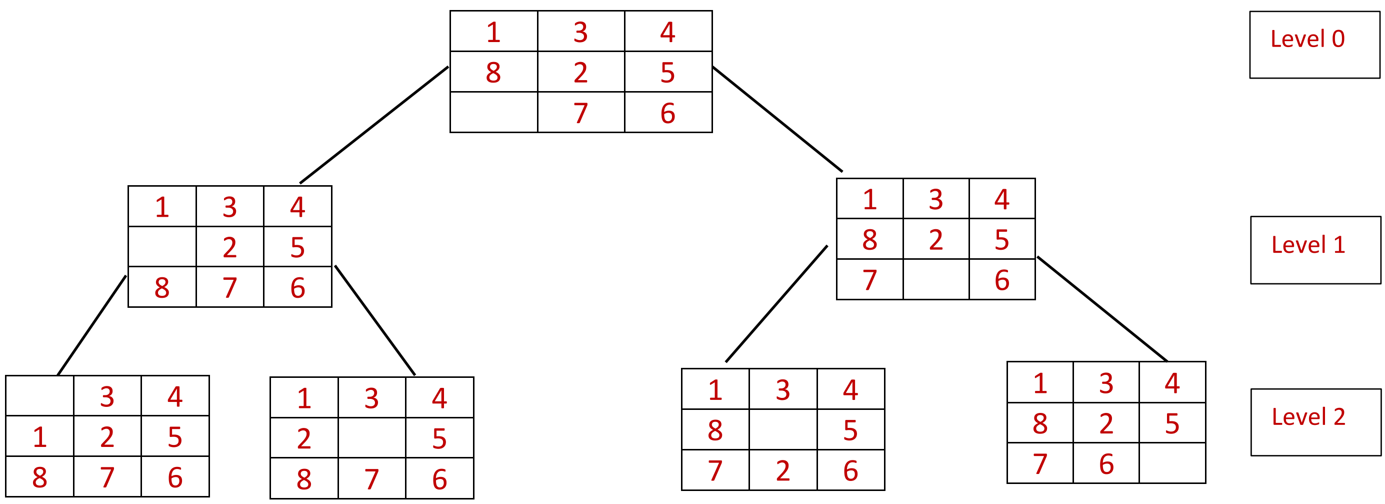
Using breadth-first search, show the search tree that would be built down to level two, assuming the root of the tree, i.e. the initial state, is level zero.

1	3	4
8	2	5
	7	6

1	2	3
8		4
7	6	5

Figure 2: 8 puzzle initial state (left) and goal state (right)

Using breadth-first search, the search tree that would be built down to level two, assuming the root of the tree:



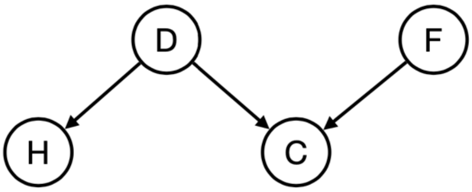


Figure 3: Bayesian network for the cake situation

- H is a random variable specifying whether there is a hole in the garden, h indicates there is a hole in the garden, ¬h indicates there is not a hole in the garden.

D	$P(D)$
d	0.3
$\neg d$	0.7

F	$P(F)$
f	0.4
$\neg f$	0.6

D	H	$P(H\|D)$
d	h	0.5
d	$\neg h$	0.5
$\neg d$	h	0.1
$\neg d$	$\neg h$	0.9

D	F	C	$P(C\|D,F)$
d	f	c	1
d	f	$\neg c$	0
d	$\neg f$	c	0.8
d	$\neg f$	$\neg c$	0.2
$\neg d$	f	c	0.9
$\neg d$	f	$\neg c$	0.1
$\neg d$	$\neg f$	c	0.1
$\neg d$	$\neg f$	$\neg c$	0.9

(i) What is the probability that there is a hole in the garden?

The probability that there is a hole in the garden:
P(H) =P(H, D) + P(H , D')
=P(H|D) * P(D) + P(H|D')*P(D')
= 0.5 * 0.3 + 0.1 * 0.7
= 0.22

(ii) What is the probability that the dog is loose, given that there is a hole in the garden?

The probability that the dog is loose, given that there is a hole in the garden:
P (D I H) = (P(H I D)*P(D))/P(H)
= (0.5*0.3)/0.22
= 0.68

(iii) What is the probability that the dog is loose, given that your cake is eaten and your flatmate is home?

The probability that the dog is loose, given that your cake is eaten, and your flatmate is home:
P(D I C,F) = P(D,C,F)/P(C,F)
= P(D)*P(F)*P(C I F, D))/(P(D)*P(F)*P(C | F, D) + P(non D)*P(F)*P(C |nonD, F)
= 0.3 * 0.4 * 1 / (0.3 * 0.4 * 1) + (0.7 * 0.4 * 0.9)
= 0.32

(iv) What is the probability that your flatmate is home, given that your cake is eaten and there is a hole in the garden?

The probability that your flatmate is home, given that your cake is eaten and there is a hole in the garden:
P(F I C,H) = (P(C)*P(H I C)*P(F I H))/(P(C)*P(H I C))
= P(F)
= 0.4

Question 7:

Consider an agent in a 2x3 gridworld. The rewards for the gridworld are given in table 2

2	-0.5	10 G
-0.5	-0.5	-0.5

Table 2: Rewards in the 2x3 gridworld

The top right cell (3, 2), marked G, is the goal state. Once the goal state is reached, the game is over. The agent starts in the bottom left cell (1, 1).

The transition model says that when the agent moves, it will go in the intended direction 80% of the time, and at right angles to the intended direction the rest of the time. So if the agent moves up, it will go up with probability 0.8, left with probability 0.1, and right with probability 0.1. Suppose the discount factor is 0.9.

7(a)
We will use value iteration to calculate the utility of each state. Assume that at iteration 0, all cells have utility 0. After one iteration, the utility of each state is equal to its reward, as in table 3. Explain why this is the case.

2	-0.5	10 G
-0.5	-0.5	-0.5

Table 3: Utilities at iteration 1

After one iteration, the utility of each state is going to be equal to its reward. This is because the utility at iteration 0 is equal to 0. Hence, after one round of value iteration, the utility of each state will be like the following according to bellman equation:

$$U_1(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|a,s) U_0(s') \quad = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|a,s) * 0 \quad (1)$$
$$= R(s) + 0 = R(s) \quad (2)$$

- U1(1, 1) = R(1, 1) + 0.9(0.8 × 0 + 0.1 × 0 + 0.1 × 0)
  = —0.5 + 0
  = —0.5
- U1(1, 2) = R(1, 2) + 0.9(0.8 × 0 + 0.1 × 0 + 0.1 × 0)
  = 2 + 0
  = 2