NTNU

Candidate Number:

EXAM

COURSE NAME: Mathematics for Games Programming

COURSE NUMBER: REA2061

EXAM DATE: 17th of August 2016

CLASS: Bachelor Games Programming

TIME: 0900-1400

COURSE CONTACTS: Bernt Tore Jensen (46250024)

NUMBER OF PAGES: 5 (inclusive front page)

AIDS ALLOWED: A: All written and printed material. All calculators.

USE A PEN FOR YOUR ANSWERS.

Remember to put your candidate number on all written pages.

When finished, separate the white and yellow page copies in two folders.

The candidate can keep the blue copy, exam questions and drafts.

All relevant calculations and explanations must be included. Read the questions carefully. All problems contribute 20 per cent towards your final grade.

Problem 1.

Given a Markov chain with states $1, 2, 3$ and 4 and matrix

$$M = \begin{pmatrix} 0.5 & 0.5 & 0.1 & 0 \\ 0 & 0.5 & 0.1 & 0.5 \\ 0 & 0 & 0.7 & 0 \\ 0.5 & 0 & 0.1 & 0.5 \end{pmatrix}$$

where the ith column gives the probabilities for leaving state i.

a)

Is the Markov chain ergodic? Is the Markov chain absorbing? Explain!

b)

i) What is the probability that the chain is in state 4 after 2 iterations if we start in state 1?

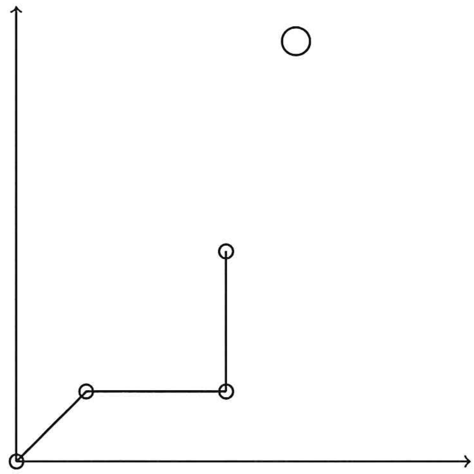
ii) What is the probability that the chain is in state 4 after 2 iterations if we start in state 3?

c)

Given that the starting state is 3, what is the probability that we are in state 4 after n iterations, where $n > 0$ is an arbitrary integer.

Problem 2.

A robot arm is fixed with a joint at the origin and two additional joints located at $(1, 1)$ and $(3, 1)$. The effector is located at $(3, 3)$. The target we want the effector to reach is located at $(4, 6)$.



a)

Calculate the lengths of the three arms and the angles at the three joints. The angle at the origin is calculated with respect to the positive x-axis.

b)

Compute the angles at two of the joints after one iteration of cyclic coordinate descent (CCD). I.e. after all the angles have been adjusted exactly once.

c)

Calculate the angles at the three joints when the effector is as close as possible to the target. Does the effector reach the target?

Problem 3.

In this problem you will use the midpoint displacement algorithm to compute height values for a horizon in 2d.

a)

Write code which does midpoint displacement on data points

```
float data[9];
```

You can assume that the noise is reduced by $1/2$ in each iteration, and you may use the predefined random number generator

```
float rnd();
```

which returns a random number in the range $-1.0...1.0$.

Note that the data should wrap around (i.e. data[0]=data[8]).

b)

Write code which scales the data so that the highest value is 1.0 and the lowest value is -1.0.

c)

Use the algorithm from a) to compute the values of

```
data[0]
data[4]
data[6]
data[7]
data[8]
```

You can use

$$0.1, -0.7, 0.3, 0.4$$

as random numbers returned from the generator in a).

Problem 4.

Given a 32 bit unsigned integer x. Write code which

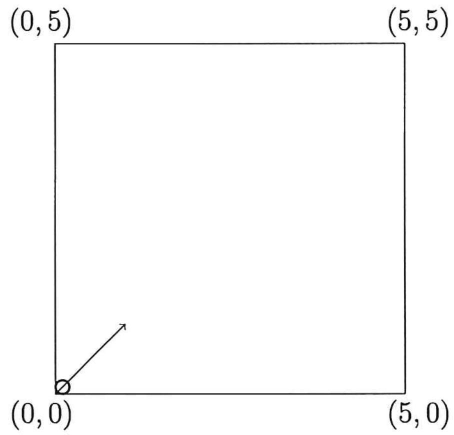
a) Counts the number of bits in x which are set to 1.

b) Counts the number of bits in x which are set to 0.

c) Counts the number of occurrences of the bit pattern 1001 in x.

Problem 5.

A tiny ball has initial velocity $(10, 10)$ when it is thrown from the lower left corner corner of a room.



a) What is the velocity when the ball hits the boundary of the room (floor, roof or walls) for the first time? Where does the ball hit the boundary?

b) What is the velocity of the ball when it hits the boundary the second time? Where does the ball hit the boundary the second time?

You can ignore air resistance and the size of the ball. The acceleration due to gravity is $(-10, 0)$ and all collisions are elastic.

All relevant calculations and explanations must be included. Read the questions carefully. All problems contribute 20 per cent towards your final grade.

Problem 1.

Given a Markov chain with states $1, 2, 3$ and 4 and matrix

$$M = \begin{pmatrix} 0.5 & 0.5 & 0.1 & 0 \\ 0 & 0.5 & 0.1 & 0.5 \\ 0 & 0 & 0.7 & 0 \\ 0.5 & 0 & 0.1 & 0.5 \end{pmatrix}$$

where the ith column gives the probabilities for leaving state i.

a)

Is the Markov chain ergodic? Is the Markov chain absorbing? Explain!

b)

i) What is the probability that the chain is in state 4 after 2 iterations if we start in state 1?

ii) What is the probability that the chain is in state 4 after 2 iterations if we start in state 3?

c)

Given that the starting state is 3, what is the probability that we are in state 4 after n iterations, where $n > 0$ is an arbitrary integer.

Answer:

a)

A chain is absorbing if any node has a path to an absorbing node. This chain is not absorbing, as it has no absorbing nodes.

A chain is ergodic if there is a path between any two states. There is no path from state 4 to state 3, so this chain is not ergodic.

b)

i) There are two paths of length 2 from state 1 to state 4.

$$1 \to 1 \to 4 \text{ and } 1 \to 4 \to 4$$

with probabilities $0.5 \cdot 0.5 = 0.25$ and $0.5 \cdot 0.5 = 0.25$. So the probability of moving from state 1 to state 4 in 2 steps is 0.5.

ii) There are three paths of length 2 from state 3 to state 4.

$$3 \to 3 \to 4, 3 \to 4 \to 4, 3 \to 1 \to 4$$

with probabilities $0.7 \cdot 0.1$, $0.1 \cdot 0.5$ and $0.1 \cdot 0.5$. So the probability of moving from state 3 to state 4 in 2 steps is $0.07 + 0.05 + 0.05 = 0.17$

This question could also be answered by doing the matrix multiplication $M \cdot M$ and reading of the relevant entries.

c)

Starting at state 3 we will eventually reach one of the states $1, 2$ or 4, with equal probability $1/3$. We observe the symmetry in the chain and so we are equally likely to end up in any of the states $1, 2$ or 4. So the probability of ending in state 4 will be $1/3$ times the probability of leaving state 3. The probability of leaving state 3 is $1 - 0.7^n$, and so the answer is $1/3 \cdot (1 - 0.7^n)$.

More formally, we could reorder the columns/rows to get the matrix

$$N = \begin{pmatrix} 0.7 & 0.0 & 0.0 & 0.0 \\ 0.1 & 0.5 & 0.5 & 0.0 \\ 0.1 & 0.0 & 0.5 & 0.5 \\ 0.1 & 0.5 & 0 & 0.5 \end{pmatrix}$$

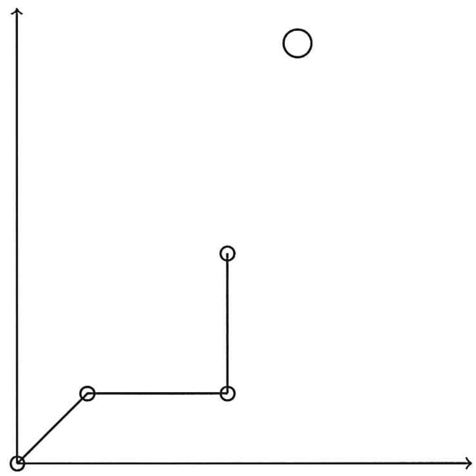
The first column in N^n is $v = \begin{pmatrix} 0.7^n \\ 0.1(1 + 0.7 + \cdots + 0.7^{n-1}) \\ 0.1(1 + 0.7 + \cdots + 0.7^{n-1}) \\ 0.1(1 + 0.7 + \cdots + 0.7^{n-1}) \end{pmatrix}$.

So the probability is $0.1(1 + 0.7 + \cdots + 0.7^{n-1})$, which is equal to the answer above using the formula for a geometric series.

Either answer gives full score.

Problem 2.

A robot arm is fixed with a joint at the origin and two additional joints located at $(1, 1)$ and $(3, 1)$. The effector is located at $(3, 3)$. The target we want the effector to reach is located at $(4, 6)$.



a)

Calculate the lengths of the three arms and the angles at the three joints. The angle at the origin is calculated with respect to the positive x-axis.

b)

Compute the angles at two of the joints after one iteration of cyclic coordinate descent (CCD) (i.e. after all the angles have been adjusted exactly once.)

c)

Calculate the angles at the three joints when the effector is as close as possible to the target. Does the effector reach the target?

Answer:

a)

The lengths are $\sqrt{2} \approx 1.414$, 2 and 2, and the angles are 45, -45 and 90 degrees.

b)

Calculations with exact and approximate values gives equal score.

After adjusting the first angle the effector is at $(3,1)+2/\sqrt{26}\cdot(1,5) \approx (3.3922, 2.9612)$. The new angle is $atan(5/1) = 78.6901$.

The length of the vector $(3.3922, 2.9612) - (1,1) = (2.3922, 1.9612)$ is 3.0934, and so after adjusting the second angle, the effector is at $(1,1) + 3.0934/\sqrt{34} \cdot (3,5) \approx (2.5915, 3.6525)$. The angle between the vectors $(3,5)$ and $(2.3922, 1.9612)$ is approx. 19.692 degrees, giving the new approximate angle $-45 + 19.692 = -25.308$ degrees.

c)

The sum of the length of the arms is approx. $1.4141+2+2 = 5.1414$, where as the target is at a distance $\sqrt{4^2+6^2} \approx 7.211$ from the origin. The effector cannot reach the target, and so the effector is the closest when the angles are $atan(4/6) = 51.34$, 0 and 0 degrees.

Problem 3.

In this problem you will use the midpoint displacement algorithm to compute height values for a horizon in 2d.

a) Write code which does midpoint displacement on data points

```
float data[9];
```

You can assume that the noise is reduced by 1/2 in each iteration, and you may use the predefined random number generator

```
float rnd();
```

which returns a random number in the range $-1.0...1.0$.

Note that the data should wrap around (i.e. data[0]=data[8]).

b) Write code which scales the data so that the highest value is 1.0 and the lowest value is -1.0.

c) Use the algorithm from *a*) to compute the values of

```
data[0]
data[4]
data[6]
data[7]
data[8]
```

You can use

$$0.1, -0.7, 0.3, 0.4$$

as random numbers returned from the generator in *a*).

Answer:

a)

```
// set initial points
data[0] = rnd();
data[8] = data[0];

int step = 4;
float alpha = 1;

while(step > 0) {
  for(int i = step; i < 8; i += 2*step) {
    data[i]=(data[i-step]+data[i+step])/2+rnd()*alpha;
  }
  step /= 2;
  alpha /= 2.0;
}
```

b)

```
float max = data[0];
float min = data[0];
for(int i = 1; i < 9; i++) {
  if(max < data[i]) max = data[i];
  if(min > data[i]) min = data[i];
```

```
}

//  Due to small dataset,
//  we could check that max > min to avoid division by zero

for(int i = 0; i < 9;  i++) {
  data[i] = ((data[i] - min) / (max - min));  // data is now 0..1
  data[i] = 2*data[i]-1; //data is now -1..1
}
```

c)

Note that the answer the students give in a) might differ from the above answers, and so consequently, the answers below could be different but still be correct. Also, using different random numbers than the ones given above (or a different order) and/or rescaling to -1...1 should not lead to a smaller score.

```
data[0]=data[8]=0.1
data[4]=(0.1+0.1)/2-0.7=-0.6
data[6]=(-0.6+0.1)/2+0.3*0.5=-0.25+0.15=0.05
data[7]=(0.05+0.1)/2+0.4*0.5*0.5=0.075+0.1=0.175
```

Problem 4.

Given a 32 bit unsigned integer x. Write code which

a) Counts the number of bits in x which are set to 1.

b) Counts the number of bits in x which are set to 0.

c) Counts the number of occurrences of the bit pattern 1001 in x.

Answers:

a)

```
int count = 0;
while(x != 0) {
  x=x&(x-1);
  count++;
}
```

```
// count is equal to the number of bits set to 1.
```

b)

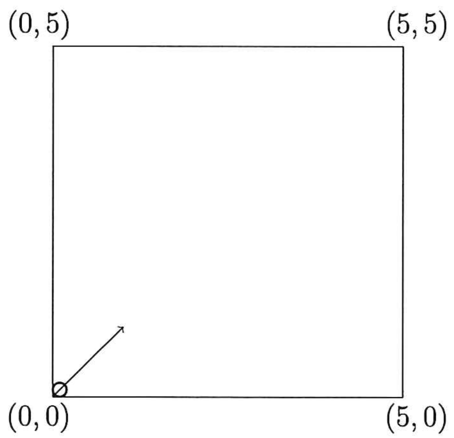
```
x = ~x;
int count = 0;
while(x != 0) {
  x=x&(x-1);
  count++;
}
// count is equal to the number of bits set to 0.
```

c)

```
int count = 0;
while(x>0) {
  if(x & 0xF == 0x9) count ++;
  x = x >> 1;
}
// count is equal to the number of patters equal to 1001
```

Problem 5.

A tiny ball has initial velocity $(10, 10)$ when it is thrown from the lower left corner corner of a room.



a) What is the velocity when the ball hits the boundary of the room (floor, roof or walls) for the first time? Where does the ball hit the boundary?

b) What is the velocity of the ball when it hits the boundary the second time? Where does the ball hit the boundary the second time?

You can ignore air resistance and the size of the ball. The acceleration due to gravity is $(-10, 0)$ and all collisions are elastic.

Answer:

a)

Veloctiy is $v = (10, 10 - 10t)$ and position is $p = (10t, 10t - 5t^2)$. The ball hits its highest point at $t = 1$ with position $(10, 5)$ which is beyond the right wall. In other words, the first collision is with the right wall, which happens at time $t = 0.5$. The position is $(5, 3.75)$ and the velocity is $(10, 5)$ at this time.

b)

If the right wall was not there, the ball would continue to its highest point at $t = 1$ and position $(10, 5)$. The velocity after the first collision is $(-10, 5)$ and so the ball hits the boundary in the upper left corner the second time (position $(0, 5)$). The velocity is $(-10, 0)$ at this time.