Unit 26 — Mathematics for IT Practitioners

Assignment 2 — Probability, Sequences, Number Systems and Statistics

Daniel Easteal

March 2017

Contents

1 Introduction

test

2 Series and Sequences and Probability

2.1 -3, 1, 5, 9, 13 . . .

2.1.1 nth term

Find a formula for the nth term of this sequence and find the 17th term using your nth term formula. Also calculate the sum of the first 17 terms of this sequence.

$$-3, 1, 5, 9, 13, \ldots$$

To start off with I will need to find the difference between the numbers in the sequence that I have been given because I believe that this is a linear progression and this will get half the answer for me. Looking at the numbers, each number is 4 more than the previous one...

$$\overbrace{-3,1}^{+4} \quad \overbrace{1,5}^{+4} \quad \overbrace{5,9}^{+4} \quad \overbrace{9,13}^{+4}$$

Now that I know the distance from each number to the next, I can then formulate the initial equation to be

$$n = 4x + ?$$

so now I just need to work out what the "?" will be. Now, for the first term the x value will be one and so we would have 4 as the value, but the first value is actually -3, so we will need to adjust to this and have the equation subtract 7 from all the values so now the equation will look like this:

$$n = 4x - 7$$

We can then test this to ensure that it works by testing the 4th value as we know the answer to that:

$$4(4) - 7 \Rightarrow 16 - 7 \Rightarrow 9$$

And this answer lines up with the answer that we are given in the question so I know that the equation is correct.

$$\therefore n^{th} term = 4x - 7$$

2.1.2 17th term

It is very simple to find the 17th term, or really any term of the sequence once you know the nth term formula, all that you have to do is replace the "x" with the term that you want and then you have to work the equation through. For this question it would go like this:

$$4x - 7 \Rightarrow 4(17) - 7 \Rightarrow 68 - 7 \Rightarrow 61$$

$$\therefore 17^{th} term = 61$$

4

2.1.3   Sum of terms

To sum the sequence that we have been given we will have to use the sum of an arithmetic sequence progression equation and this looks like this:

$$\frac{n(a_1 + a_n)}{2}$$

this sequence sequence is an arithmetic sequence because it advances in the series by adding numbers rather than multiplying by them. In this equation, all of the symbols start for something and I will explain them below:

n = represents the number of terms that you want to add, in this case it is 17

a_1 = represents the initial number of the sequence, in this case it is -3

a_n = represents the last number of the sequence, in this case this is 61 as we calculated that before

Now I just need to place these numbers where their letter counterpart as in the equation and then work through the equation to get the answer:

$$\frac{17(-3 + 61)}{2} \Rightarrow \frac{17(58)}{2} \Rightarrow \frac{986}{2} \Rightarrow 493$$

$$\therefore \text{ the sum of the first 17 terms of the equation 4x-7 is 493}$$

2.2   81, -27, 9, -3 . . .

2.2.1   nth term

As with before I will start off with working out the difference between the values in the list that I have been given.

$$\overbrace{81, -27}^{108} \quad \overbrace{-27, 9}^{36} \quad \overbrace{9, -3}^{12}$$

As you can see here, the numbers do not match up and they are not the same, this then presents a problem because instead of adding to get to the next number we are now multiplying, this can also be shown as the difference between the difference of all the numbers is 3. What I mean by this is that the above values that are 108, 38 and 12, you will that to get to the next number you would need to divide by 3, this shows that this is a geometric progression and that I will have to use another equation.

Looking at the numbers you can see that to get to the next number in the series you would divide by -3. 3 because that is the way the differences work and then the - because the sign of the series changes every other number. Now that we know the multiple of the series we need to work out the offset.

Again like last time the we write it out in a standard form and this would be:

$$a \times r^{n-1}$$

Now that we know "r" (the common ratio) to be -1/3 we just need to put in "a" as the first number and we know that that is 81, to then it would look like this:

$$a \times r^{n-1} \Rightarrow 81 \times \frac{1}{3}^{(n-1)}$$

2.2.2 10th term

This section is again very easy as all that I have to do is put 10 into the nth term formula and then I well get the answer that I want. This would then work through like so:

$$81 \times -\frac{1}{3}^{10-1} \Rightarrow 81 \times -\frac{1}{3}^{9} \Rightarrow 81 \times -\frac{1}{19683} \Rightarrow -\frac{81}{19683} \Rightarrow -\frac{1}{243}$$

2.2.3 Sum to the 5th element

In order to sum a geometric series we need to know a new formula and this is the formula that we need:

$$S_n = \frac{a_1(r^n - 1)}{r - 1}$$

As the other ones I will just insert the numbers into the formula and then go about and solve the equasion to get to the answer that we need to get to:

$$\frac{a_1(r^n - 1)}{r - 1} \Rightarrow \frac{81(-\frac{1}{3}^{5} - 1)}{-\frac{1}{3} - 1} \Rightarrow \frac{81(-0.004115226... - 1)}{-1.\dot{3}} \Rightarrow \frac{81 \times -1.004115226...}{-1.\dot{3}} \Rightarrow \frac{-81.\dot{3}}{-1.\dot{3}} \Rightarrow 61$$

2.2.4 Sum to infinity

In order to sum a geometric series to infinity you need to ensure that the common ration is between -1 and 1 as otherwise the sum would be infinity and not calculated. The new equation that we need to sum to infinity is as follows:

$$S = \frac{a}{(1 - r)}$$

As in the previous examples, the "a" represents the first number in the series and the "r" represents the common ration. In this case that would be 81 and -1/3 so now the formula would look like this:

$$S = \frac{a}{(1 - r)} \Rightarrow \frac{81}{(1 - -\frac{1}{3})} \Rightarrow \frac{81}{(1 + \frac{1}{3})} \Rightarrow \frac{81}{(1.\dot{3})} \Rightarrow 60.75$$

2.3 3r - 2r^2 + r^3

In this section I have to calculate the answer to the following:

$$\sum_{r=1}^{6}(3r - 2r^2 + r^3)$$

The way that this works is that I have to calculate the part of the equation that is inside the brackets by replacing the "r" in this case each time from the value on the bottom to the vale on the top. Then you add all these answers and you are done.

$$((3\times1)-(2\times(1^2))+(1^3)+((3\times2)-(2\times(2^2))+(2^3)+((3\times3)-(2\times(3^2))+(3^3)+((3\times4)-(2\times(4^2))+(4^3)$$

$$+((3 \times 5) - (2 \times (5^2)) + (5^3) + ((3 \times 6) - (2 \times (6^2)) + (6^3)$$

$$3 - 2 + 1 + 6 - 8 + 16 + 9 - 18 + 27 + 12 - 32 + 64 + 15 - 50 + 125 + 18 - 72 + 216 \Rightarrow 322$$

$$\therefore \sum_{r=1}^{6}(3r - 2r^2 + r^3) = 322$$

6

2.4 5 balls in a bag

There are 3 red and 2 yellow balls in a bag:

2.4.1 yellow ball

The probability that a yellow ball is selected is 2/5 due to the fact that there are 2 yellow balls in the bag out of a total of 5. This can be written as 0.4

2.4.2 2 yellow balls in a row

When there are 2 chances that need to be added up in a row then the individual chances are multiplied. For this we would then multiply the 0.4 chance from the previous question by itself as it is the same question we just have to do it twice in a row.

$$0.4 * 0.4 = 0.16e$$

$\therefore$ the chance of getting two yellow balls in a row is 0.16 or $\dfrac{2}{25}$
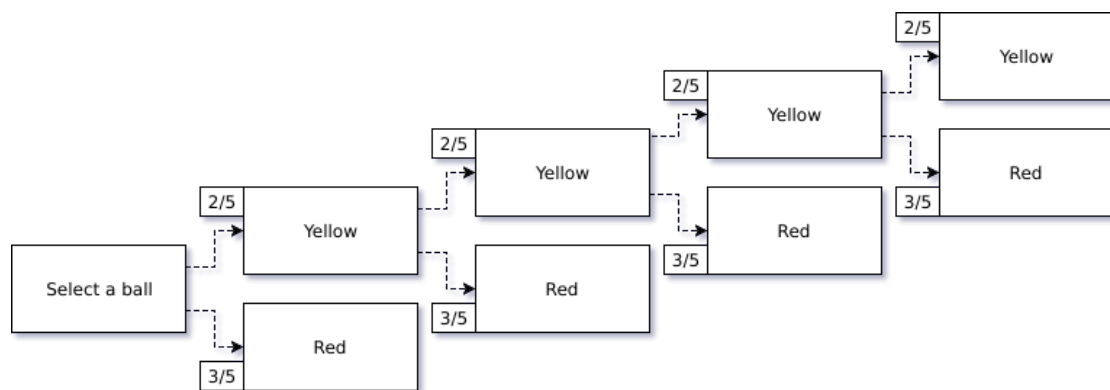
2.4.3 Tree

Figure 1: a tree diagram

2.5 In a year group

2.5.1 Venn diagram

This section will be on the insert provided with the document...

2.5.2 Only Computer science

To start to answer this question I first need to add up how many different students there are in total and then I can work out the probability.

$$70 + 83 + 15 + 12 + 10 = 190$$

Now that I know there are 190 students in the year I can now look at the information and see that there are 70 students that fit the category of only studying computer science and to the probability is now:

$$70/190 \Rightarrow 7/19 \Rightarrow (100/19) + 7 = 12.26315789473684210526\%$$

$$\text{or a probability of } 7/19 = 0.36842105263157894736$$

2.5.3   Engineering

looking at the information that I have there are 83+12 students that study Engineering in one way of another and as such there is a probability as follows for one to be randomly selected:

$$95/190 \Rightarrow 1/2 \text{ That is } 50\% \text{ or a probability of } 0.5$$

2.6   Betting game

This section will be on the insert provided with the document...

3   Number Systems

3.1   Binary, Octal, Decimal and Hexadecimal conversion

	Denary	Binary	Octal	Hexadecimal
a	**22**	10110	26	**16**
b	11	1011	**13**	B
c	**41**	101001	51	29
d	20	**10100**	24	14
e	30	11110	**36**	1E
f	42	101010	52	**2A**
g	**271**	100001111	417	10F
h	44	101100	54	2C
i	62	111110	76	3E

4   Number Systems calculations

4.1   a + f in hexadecimal

$$22 + 42 = 64 \rightarrow 64_{10} = 40_{16}$$

$$\therefore A + F = 40$$

4.2   g * f in hexadecimal

$$271 * 42 = 11382 \rightarrow 11382_{10} = 2C76_{16}$$

$$\therefore g * f = 2C76$$

4.3 f - b in hexadecimal

$$42 - 11 = 31 \rightarrow 31_{10} = 1F_{16}$$

$$\therefore f - b = 1F$$

4.4 a + e in Octal

$$22 + 30 = 5252_{10} = 64_8$$

$$\therefore a + e = 64$$

4.5 e - b in Octal

$$30 - 11 = 1919_{10} = 23_8$$

$$\therefore e - b = 23$$

4.6 a + d in binary

$$22 + 20 = 42 \rightarrow 42_{10} = 101010_2$$

$$\therefore a + d = 101010$$

4.7 a * d in binary

$$22 * 20 = 440 \rightarrow 440_{10} = 110111000_2$$

$$\therefore a * d = 110111000$$

4.8 h + i in binary

$$44 + 62 = 106 \rightarrow 106_{10} = 1101010_2$$

$$\therefore h + i = 1101010$$

4.9 h * i in hexadecimal

$$44 * 62 = 27282728_{10} = AA8_{16}$$

$$\therefore h * i = AA8$$

5 Data task

6 Recursion

For this section I will go through and explaining how the binary search algorithm works and it fits in with the idea of recursion. Below you will see a basic format for the algorithm written in pseudo code:

```
for every item in the data set, read it into an array.
if the array is not sorted then sort it
start:
go to the middle of the array
is the current item what you are looking for?
....no: is the current item less then the item you are looking for?
........yes: remove all items in the array that are lower and return to start.
........no: remove all items in the array that are greater and return to start.
....yes: you are done as you  have found the item you are looking for, end the program.
```

As you can see here there are not many stages that are needed for the search algorithm and it is very simple to do. Due to this, the whole part of the program after start: goto will be run over and over again and this will lead to the item that you wand being found. I will now go through the algorithm step by step so that you know why each part is done like it is.

```
for every item in the data set, read it into an array.
```

This line will read all of the information that you need into the program so that you can actually modify it and run the search.

```
if the array is not sorted then sort it
```

This line will sort the data in the array so that it is in an order that can be compared easily, like a cost or number. This will mean that later when you test if the data is greater or lower, you will know that you are removing only data that will not be correct.

```
start:
```

This line just marks this section in the code with a name so that I cat return to it later in the program.

```
go to the middle of the array
```

This line will set the location of the program to look at the middle of the array of numbers, if the length of the array is odd then the program should just go to either one of the middle values as it does not really matter.

```
is the current item what you are looking for?
```

This will check if the item in the array that you are currently looking at in the program is actually the one that you want. This well then lead off to two conditions depending on if you have found the value that you are looking for or not. If you have found the item that you are looking for then you are done, but otherwise the program carries on.

```
....no: is the current item less then the item you are looking for?
```

This line will run if the item that the program is on is not the value that you want, that is what the no: part means. Now this is where the clever part of the algorithm starts. If the value of the item that you want is lower then where you are at the moment then you can remove all of the items that are greater than the current item as it is less then this one so it is not higher. This is why the array was sorted to start off with as now you can just remove the half of the array that is higher, and just like that you have halved the values to search.

```
........yes: remove all items in the array that are lower and return to start.
```

If the item that you are are on is lower than what you want then you can remove all of the items that are lower and this is what this section does. After you have removed the values then you can return to the start: marker of the program and recur it again.

```
........no: remove all items in the array that are greater and return to start.
```

If the item that you are are on is greater than what you want then you can remove all of the items that are greater and this is what this section does. After you have removed the values then you can return to the start: marker of the program and recur it again.

```
....yes: you are done as you  have found the item you are looking for, end the program.
```

If the value that you have at the middle of the selection is the number that you are looking for then you have found it and can see what information it has that you need. Now you just get that information and save it and end the program and you are done. This is the terminal condition.

7 Use of Number Systems

In this section I will go through how binary, octal and hexadecimal are used and are applied in areas of computing with how they are used and how they work. For this I will be showing how binary is used with ASCII, how hexadecimal is used with MIME and how Octal is used with Unix file permissions.

7.1 binary - ASCII

One way that binary is used in computer science is in the use of ASCII. ASCII stands for the American Standard Code for Information Interchange and this is the standard that is used for how the standard set of English characters and numbers are encoded in binary on the computer and how it deals with them. With ASCII the letter "a" for example will have a value inside the computer of 1100001. So whenever the computer does anything with the letter a it will use this value of 1100001 internally. Now a cl;aver part of this is that "b" is 1100010, one more than "a" but the right most digits correlate to the numbers as 1 is 1 and 10 is 2. So for example if you had "s" the 19th letter in the alphabet then its ASCII value would be 1110011, and 19 in binary is 10011. This ASCII also extends to capital letters in a similar fashion as well, just with a different prefix and there is also all of the things that you can type on a English keyboard.

7.2 Hexadecimal - MIME

To start off with, mime is a mechanism that enables the transmission of information in email format between computers that may not have the same language. Due to the fact that the whole point is that the computers have different languages, the 8 bits that where used in the previous example will not be enough and as such we need to use 2 Hexadecimal bits, in addition to this it also uses formatting to ensure that all messages fit a certain line length so that it is compatible and so on. This as you can see will use hexadecimal characters to represent the information and this is how they are used in computing. So, for example, a value of 12 will be "=0C".

7.3 Octal - Unix file permissions

A way that octal is used in computing is in the UNIX and UNIX-like operating systems for setting file permissions for the users of the system. The way that the file permissions are handled in UNIX is that the file has meta data set aside for it that will store this information and there are 9 main bits for the file access permissions and a few more for the file type. The 9 main bits are separated into 3 sections that represent the access that the owner, group and anyone else has for the file. These 3 bits then represent the ability that they have to read, write and execute the file. So the bits are arranged like so

`rwxrwxrwx`

As you can see, there are the 3 groups there and the (r)ead, (w)rite and e(x)ecute that the 3 groups have. Now the octal comes in when you want to change the value of these bits and the way that you do this is through the use of the chmod command. You give the chmod command the file that you want to edit and then you give it 3 octal numbers. The 3 octal numbers are then converted to binary and this then lines up with bits in the file and where there is a one that will be active. Lets say that you have a file called .xinitrc that you wanted to change the permissions of and you ran the command chmod .xinitrc 752, lets see what would happen:

$$7_{10} = 111_2 \quad \& \quad 5_{10} = 101_2 \quad \& \quad 2_{10} = 010_2$$

$$\frac{\downarrow 111101010 \downarrow}{\text{rwxrwxrwx}} \Rightarrow \text{rwxr-x-w-}$$

And now the permissions are changed using just octal numbers. This process is called masking.

8 Network Planning

In this section I will go through and design 3 subnets for 3 different networks each of a different size.

8.1 1000 hosts

When designing a subnet you will need to find the power of 2 that is as low as possible while being higher than the target number of hosts. In this case the number is 1000, and the power of 2 that satisfies that requirement is 2^{10} as that is 1024. Now that I have this number I will need to then take a string of 32 ones and make, in this case the 10 rightmost ones 0's this is also known as /22 as the first 22 digits are 1's. This will mean that I now have the following:

$$11111111111111111111111111111111 \rightarrow 11111111111111111111110000000000$$

Now I need to separate that number into four groups of 8 bits like so and then convert those 8 bet numbers into decimal to get the subnet mask and address:

$$\frac{\downarrow 11111111 \quad 11111111 \quad 11111100 \quad 00000000 \downarrow}{255 \qquad\quad 255 \qquad\quad 252 \qquad\quad 000} \Rightarrow 255.255.252.0$$

Now I have the subnet mask 255.255.252.0. And in this case this would be the most efficient and justified method for this network due to the fact that it allows for all of the devices that need to connect to the network to connect while taking up as few of the bits for the subnetting as possible so that more of these networks will be able to be up at a time. Another thing to note is that the network id would be 192.168.0.0. In addition to this, the broadcast would be

192.168.3.255 as that is the highest ip that is available on this network. This would mean that the any host on the network would get an ip address between these two and so there would be 1022 possible addresses for the hosts.

8.2 200 hosts

with this I will do the same process but with the number 200 this time.

The lowest number that satisfies what I need is 2^8 as this number is 256, this is known as /24 as the first 24 bits are on. Now I will have the binary number as follows:

$$11111111111111111111111111111111 \rightarrow 111111111111111111111111100000000$$

This then goes to:

$$\frac{\downarrow 11111111 \quad 11111111 \quad 11111111 \quad 00000000 \downarrow}{255 \qquad 255 \qquad 255 \qquad 000} \Rightarrow 255.255.255.0$$

And so the address is 255.255.255.0. And as before this is efficient for the same reason. The network ID would be 192.168.0.0, the broadcast would be 192.168.0.255, this would leave 254 ip addresses between these numbers for the hosts.

8.3 30 hosts

with 30 hosts the best number would be 2^5 and that is 32 as that fits in all the hosts and this is /27. This would lead to the binary of this:

$$11111111111111111111111111111111 \rightarrow 111111111111111111111111111100000$$

$$\frac{\downarrow 11111111 \quad 11111111 \quad 11111111 \quad 11100000 \downarrow}{255 \qquad 255 \qquad 255 \qquad 224} \Rightarrow 255.255.255.224$$

And to the address is 255.255.255.224, the ID is 192.168.0.0, the broadcast is 192.168.0.32, this would leave the perfect 30 places for the hosts on the network between these values. If this network was to ever expand then they would need to use one more bit as a 0 in the mask.

8.4 ipv4 vs ipv6

ipv4 and ipv6 are both ways to identify computers on a network and how they should be accessed and sent information. ipv6 is the upgraded version of the common and thoroughly used ipv4. Before I can explain the differences you need to know how they work and what they do. So, an ipv4 address will look like this: "192.168.0.1" as you can see, this is a collection of four numbers that can each go up to a total of 255 and they are all separated by a dot. This would mean that there would be a maximum of 255*255*255*255 different numbers that could happen and that is a value of 4,228,250,625. Now an ipv6 address looks like this: "2001:db8:85a3:8d3:1319:8a2e:370:7348" as you can see, this is a collection of 8 sections of 4 hexadecimal numbers each separated by a colon. Due to the fact that each block is four hexadecimal numbers the different number of possibilities are 65536, this would then go to 65526^8 different possibilities and that is a huge 340282366920938463463374607431768211456. And here is the main reason that ipv6 exists, just the amount of addresses that it provides is so high. And even though the ipv4 number was big we have used all of those and so we needed to expand.

8.5 Network classes

When you are talking about networks, there are a few different classes for networks that are intended for different types of networks. The types of networks are called class A, B and C. These are distinguished by the subnet mark that they use.

$Class$	$FirstID$	$LastID$
$ClassA$	1.0.0.0	126.0.0.0
$ClassB$	128.0.0.0	191.255.0.0
$ClassC$	192.0.0.0	223.255.255.0

From this you can see that the Last ID will be the same as the subnet mask for the network and then the first ID will be the first ip address that will be available. As you can see, the class A networks will have the ability to have many, many devices connected at the same time, but not many of then can be connected. Due to this there networks are used by ISP's because that is what they need. On the other end of the scale you have the Class C networks and these cannot have too many devices connected to them but there can be a lot of them. Due to this this would be suited to the home owners that the ISP will serve, and that is what happens. All the classes in-between are used by companies of different sizes and so on.

8.6 Classless Inter Domain Routing (CIDR)

The final part on this networking section is the CIDR, this stand for Classless Inter Domain Routing and is a way of allow there to be more ip addresser allocated to devices while still keeping the existing infrastructure in place. If you look at how subnetting allows there to be more networks connected together which in turn adds in more ip addresses to be able to be used in there own networks. This is how supernetting or CIDR works but instead you perform the action on the subnets which in turn performs the process on the ip addresses. Using this as you can see, you can get many more subnets and in turn many more ip addresses and hosts connected to the network.