1. Examine the following formal descriptions of sets so that you understand which members they contain. Write a short informal English description of each set.
   1. {1, 3, 5, 7, ... }

This set contains the set of odd natural numbers.

* 1. { ... , -4, -2, 0, 2, 4, ... }

This set contains all even integers.

* 1. {n l n = 2m for some m in N}

This set contains all Natural numbers divisible by 2.

* 1. {n l n = 2m for some m in N, and n = 3k for some k in N }

This set contains Natural numbers divisible by both 2 and 3.

* 1. { n l n is an integer and n = n + 1}

This set contains all integers that are equal to 1 added to that number.

1. Write formal descriptions of the following sets
   1. The set containing the numbers 1, 10, and 100

{ n | n is a Natural number and n = 1,10,…,100 }

* 1. The set containing all integers that are greater than 5

{ n | n is an Integer and n > 5 }

* 1. The set containing all natural numbers that are less than 5

{n | n is a Natural number and n < 5 }

* 1. The set containing nothing at all

{ ∅ } or NULL set

1. Let A be the set *{x, y, z}* and B be the set *{x, y }.*
   1. Is A a subset of B?

No

* 1. Is B a subset of A?

Yes

* 1. What is A ∪ B?

{x, y ,z}

* 1. What is A ∩ B?

{x, y}

* 1. What is A × B?

A x B = { (x, x), (x, y), (y, x), (y, y), (z, x), (z, y) }

* 1. What is the power set of B?

P(B) = { (∅), (x), (y), (x, y) }

1. If A has ***a*** elements and B has ***b*** elements, how many elements are in A x B?

Explain your answer.

AxB = There are a\*b elements.

A = {1,2,3,…,a}

B = {1,2,3,…,b}

{(1,1), (1,2), (1,3), …(1,b), (2,1), (2,2), (2,3), …(2, b), (a,1), (a,2), (a,3), …(a,b)}

1. If C is a set with ***c*** elements, how many elements are in the power set of C? Explain your answer.

P(C) = { 2^c, where c is the number of elements in the set C }

|  |  |
| --- | --- |
| n | f(n) |
| 1 | 6 |
| 2 | 7 |
| 3 | 6 |
| 4 | 7 |
| 5 | 6 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| g | 6 | 7 | 8 | 9 | 10 |
| 1 | 10 | 10 | 10 | 10 | 10 |
| 2 | 7 | 8 | 9 | 10 | 6 |
| 3 | 7 | 7 | 8 | 8 | 9 |
| 4 | 9 | 8 | 7 | 6 | 10 |
| 5 | 6 | 6 | 6 | 6 | 6 |

1. Let X be the set {1, 2, 3, 4, 5} and Y be the set {6, 7, 8, 9, 10}.

The unary function *f: X****→*** *Y* and the binary function *g: X × Y****→*** *Y* are described in the following tables.

* 1. What is the value of f(2)?

7

* 1. What are the range and domain of *f* ?

Range = {6, 7}

Domain = {1, 5}

* 1. What is the value of *g(2, 10)*?

6

* 1. What are the range and domain of *g*?

Range = {6, 10}

Domain = {1, 5}

* 1. What is the value of *g(4, f(4) )*?

G(4, f(4)) = g(4, 7) = 8

1. For each part, give an example of a relation that satisfies the condition.
   1. Reflexive and symmetric but not transitive

A = {1, 2, 3, 4}

(1, 1), (2, 2), (3, 3), (4, 4), (1, 3), (3, 1), (2, 4), (4, 2)

* 1. Reflexive and transitive but not symmetric

A = {1, 2, 3, 4}

It is reflexive because 1 <= 1 is TRUE

It is transitive because 1 <= 2 and 2<= 3 means that 1 <= 3 is TRUE

But it is not symmetric because 1 <= 2 does not mean that 2 <= 1.

1. The time complexity of algorithms can be categorised as, for example: *cubic, logarithmic, exponential, constant, linear, factorial, loglinear, quadratic****.***

Write these functions out in order of fastest to slowest, giving their big-O notation, and for each give an example of an algorithm that exhibits this growth pattern

• Constant -- O(1)

E.g. x = y + 1

• Logarithmic -- O(log n)

E.g. while (x>1)

{x = x/2}

• Linear -- O(n)

E.g. for (i=0; i<x; i++)

{y+=1}

• Log-Linear -- O(n log n)

E.g. A sorting algorithm such as merge sort

• Quadratic -- O(n^2) // n to the power of 2

E.g. for (i=0; i<x; i++) {

For (j=0; j<x; j++) {

a+=1;

}

}

• Cubic -- O(n^3) // n to the power of 3

E.g. for (i=0; i<x; i++) {

For (j=0; j<x; y++) {

For (z=0; z<x; z++) {

A+=1;

}

}

}

• Exponential -- O(2^n) // 2 to the power of n

An example of exponential growth would be a Fibonacci sequence.

• Factorial -- O(n!)

X! = x\*(x-1)!

Factorial(x) = x \* factorial(x-1)

1. Using the formal definition of Big-O, show that:
2. T(n) = 4n + 5 is O(n)

= 4n + 5 <= 4n + 5n

= 4n + 5 <= 9n for all n >= 1 (c = 9 and n0 = 1)

1. T(n) = n3 +3n2 - 6 is O(n3)

= n3 +3n2 – 6 <= n3 +3n2

= n3 +3n2 – 6 <= n3 +3n3

= n3 +3n2 – 6 <= 4n2 for all n >= 0 (c = 4 and n0 = 0)

1. T(n) = log2 n3 is O(log n)

= log2 n3 for all n <= 3 log2 n for all n >= 3( c = 3 and n0 = 1)

1. T(n) = (3n + 5)\* log2 n is O(nlog n)

= 3nlog2 n + 5 log2 n <= 3nlog2 n + 5 log2 n

= 3nlog2 n + 5 log2 n <= 3nlog2 n + 5nlog2 n

= 3nlog2 n + 5 log2 n <= 8nlog2 n for all n >= 8 (c = 8 and n0 = 1)