Theta Notation

We formally introduce theta notation, which forms the basis of mathematical analysis of algorithms.

Example

Let's consider the following three functions:

$$f(n)=n^2+2$$
 $f(n)=2n^2-1$ $f(n)=2n^2+4$

The table below shows how these functions grow as the value of n grows from 0 and onwards. Note that we aren't considering negative values for \mathbf{n} , but we'll explain why shortly.

f(n)	$n^2 + 2$	2n ² - 1	$2n^2 + 4$
f(0)	2	-1	4
f(1)	3	1	6
f(2)	6	7	12
f(3)	11	17	22
f(4)	18	31	36
f(5)	27	49	54

f(6)	38	71	76	
f(7)	51	97	102	
f(8)	66	127	132	
f(9)	83	161	166	
f(10)	102	199	204	

We can observe from the output that the function $f(n) = 2n^2 - 1$ grows faster than the function $f(n) = n^2 + 2$, but it grows slower than $f(n) = 2n^2 + 4$ once n becomes greater than or equal to 2. The astute reader would notice that the output of the function $f(n) = 2n^2$ is, in a sense, being **sandwiched** by the output of the other two functions when $n \ge 2$. All the yellow rows in the above table, show the values of the function $f(n) = 2n^2 - 1$ being sandwiched by the output of the other two functions.

Formal Definition

Whenever we can bound the output of a function f(n) by the outputs of two other functions in the following form, we say that the function f(n) belongs to the set $\Theta(g(n))$ (pronounced as theta of g(n))

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$$

where $n > n_0$ and c_1 and c_2 are positive

Let's apply the formal definition to the example functions we just discussed above.

$$n^2+2 < 2n^2-1 < 2n^2+4$$
 $1(n^2+2) < 2n^2-1 < 2(n^2+2)$ $where \ n \geq 2 \ and \ c_1=1 \ c_2=2 \ and \ g(n)=n^2+2$

We can choose positive constants so that our function $2n^2$ - 1 becomes sandwiched by the other two functions as soon as n becomes greater than 2. Hence we can say that the function $2n^2$ - 1 is Θ of g(n) where g(n) equals n^2 + 2.

Explanation

There are several points to ponder about the above relationship.

• Note that to satisfy the formal definition, none of the three functions can produce negative values after n has crossed a certain threshhold marked by n_0 . In our example, f(n) produces a negative value when n=1. However, once $n \ge 2$ all subsequent values produced by all three functions are nonnegative. Consider the following function.

$$f(n) = (-1)^n n^2$$

The above function will not produce nonnegative values (i.e. 0 or positive numbers) for any threshold that n crosses, as it will oscillate between a negative and a positive number for odd and even values of n respectively. We require g(n) to be *asymptotically nonnegative*, a mathematically-fancy term meaning that as n grows and passes a certain value n_0 , it should always produce nonnegative numbers as output. This restriction is partly because it makes it easier to reason about the complexity of algorithms, as input is either none (i.e. 0 or some positive number).

• Once our function f (n) satisfies the inequality (i.e. we are able to sandwich it) we can claim that,

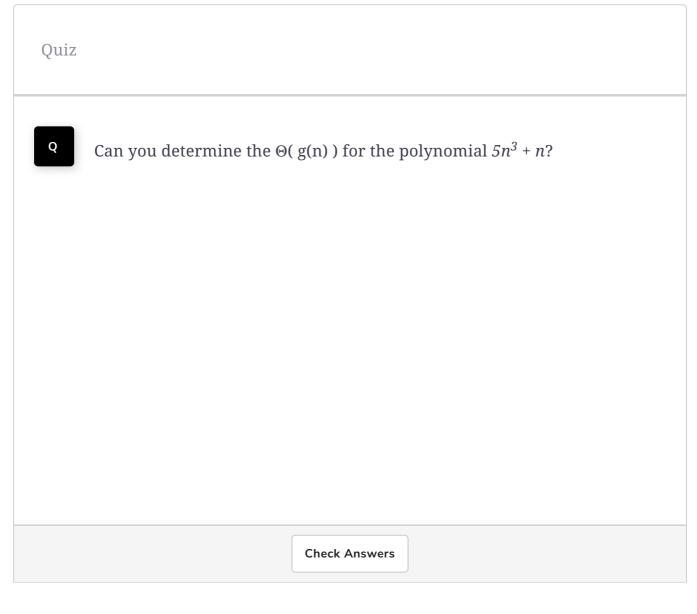
$$f(n) = \Theta g(n)$$

Note the use of equality instead of membership. Even though Θ g (n) is a set and f (n) belongs to the set g (n), we take the liberty to use the

equality instead of membership when undertaking algorithmic analysis.

• We say that g(n) is a *asymtotically tight bound* for f(n), which is just a mathematical way of saying that the growth of f(n), as n grows larger and larger, mirrors that of g(n). Or f(n) is equal to g(n) within a constant factor when the values of n are past the threshold n_0 .

• Note that in our example we used constants $c_1 = 1$ and $c_2 = 2$, but we could use any other set of constants that satisfy the inequality.



 $\Theta(1)$

Let's say we have a function that always outputs the same constant value for any value of n.

$$f(n) = 7$$

The above function will always produce 7, no matter what the value of n is. Let's try to see if we can find a $\Theta(g(n))$ for our constant function.

$$egin{aligned} c_1 n^0 < f(n) < c_2 n^0 \ & c_1 n^0 < 7 < c_2 n^0 \ & c_1 < f(n) < c_2 \end{aligned}$$

We can now choose c_1 = 2 or c_2 = 8 and claim that our function f(n) belongs to the set $\Theta(n^0)$, which is equivalent to $\Theta(1)$ as we know n^0 =1. This is considered slight abuse of the notation, as the expression $\Theta(1)$ doesn't indicate which variable is tending to infinity.

 $\Theta(1)$ is used to denote either a constant or a constant function with respect to some variable.