Nem Negash

1. For all n > 5, and let n represent the function 3x+4y = n.

The base case is as follows.

Base Case:

Given that n = 9, x = 3, and y = 0, then 3(x) + 4(y) = 3(3) + 4 (0) = 6 which = n.

Thus, it is shown that n = 9.

Now, suppose that n = 3x + 4y for some integers x and y >= 0.

We must show that n+1 = 3x + 4y + 1, where x and y >= 0.

Case 1:

Given that n = 3x + 4y, suppose that x >= 1 and y >= 0.

Here, we must subtract one 3 pack of biscuits from our total and and one 4 pack of biscuits.

Now,

n-x + y = 3(x-1) + 4(y + 1);

n-x + y = 3x - 3 + 4y + 4;

n-x + y = 3x + 4y + 1.

Case 2:

Given that n = k+1. Suppose n = 3x+4y where y >= 2.

Here, we must subtract two 4 packs of biscuits from our total and add three 3 packs.

Now,

k+1 = 3(x+3) + 4(y-2);

k+1 = 3x + 9 + 4y - 8;

k+1 = 3x + 4y + 1;

n = n +1.

Using induction in Case 1 and Case 2 above that we can purchase any

number of biscuits, n, where n &gt; 5, as a combination of 3 and 4 packs of biscuits.

1. It is given that if T(N) = O(f(n)). Thus, this implies that C > 0 and no > 0. Then, T(n) <= C1f(n) for the value of n>= n1 . Additionally, e(n) = O(g(n)), which implies that C2 > 0 and n2 >0. Then, e(n) < C2 g(n) for the value of n >= n2. It can then be stated that T(n) < Cg(n). Thus, T(n) = O(g(n)). So, it can be stated that the run time of T(N) and f(n) is O(g(n)) for all values of n >= maximum(n1 ,n2).
2. This is possible because f(n) is O(g(n)) when there exists some constants, c and n0 where f(n) less than or equal to c \* g(n) for all n >n0 . It is also possible for O(n2) to be faster than O(n lg3 ) for n < 100 since at less than 100, the input n is not large enough for O(n lg3 ) to be faster than 0(n 2 ).
3. Let A be an array with n integers in it all in ascending form and t is the value we are trying to reach and let a and b be the two integers to hold the indices and they both start at 0;

int Findpair(A,t,a,b){

if(b == A.size()-1 && a == A.size()-1){return 0;}//base case if there is no pair

if(b < A.size()-1){b++}//move to the next index

else{a++; b = 0}//if b has reached the last indx

if(A[a]+A[b] == t){pair<int,int> ret(a,b); return ret}//if pair is found

else{Findpair(A,t,a,b)}//recursive call

}