If I modified the function to be:

x = f(n)

  x = 1;

  y = 1;

  for i = 1:n

     for j = 1:n

       x = x + 1;

       y = i + j;

4. Will this increate how long it takes the algorithm to run (e.x. you are timing the function like in #2)?

Yes, compared to the original form of the algorithm, the modification to the function will probably result in an increase in algorithm execution time. You've added an additional action to the nested loops in the changed function:

Despite being a straightforward arithmetic addition, this operation is performed ($n * n = n^2$) times since it is contained within two nested loops.

Adding this extra operation will increase the time complexity of the algorithm, making it run slower compared to the original version where only one operation (incrementing x by 1) was being performed within the nested loops.

So, yes, if you are timing the modified function similarly to how you timed the original function (#2), you should expect it to take longer to execute due to the additional operation.