Hands-On 3

1. For the given function the runtime would be calculated with the following summation formula:

|  |  |
| --- | --- |
| n | T(n) |
| 1 | 7 |
| 2 | 16 |
| 3 | 29 |
| 4 | 46 |
| 5 | 67 |
| 6 | 92 |
| 7 | 121 |
| 8 | 154 |
| 9 | 191 |
| 10 | 232 |
| 100 | 20302 |
| 1000 | 2003002 |
| 10000 | 200030002 |



1. For this case, I pick c1 = ½ and c2 = 5, because: (1)  
     
   Αnd we derive the following notations:
2. n\_0 in our case is chosen to be equal to 2 since for any n ≥ 2, inequality (1) is valid  
   ![A graph of a function

   Description automatically generated]()  
   As shown in the figure above, at x=1.4 the upper bound is larger than T(n), therefore, n\_0=2 is the first integer that T(n) respects both bounds
3. In case the function changed to:   
   x = f(n)

   x = 1;

   y = 1;

   for i = 1:n

        for j = 1:n

             x = x + 1;

        y = i + j;  
The new T(n) would be:

1. The results will be as follows in the below table:

|  |  |
| --- | --- |
| n | T(n) |
| 1 | 10 |
| 2 | 23 |
| 3 | 42 |
| 4 | 67 |
| 5 | 98 |
| 6 | 135 |
| 7 | 178 |
| 8 | 227 |
| 9 | 282 |
| 10 | 343 |
| 100 | 30403 |
| 1000 | 3004003 |
| 10000 | 300040003 |

In the low n’s there is a difference, but as n gets larger the difference to the first function is almost negligible.