Hands on 3

① - The initialization step takes constant time that means $O(1)$

- The outer loop runs n times and the inner loop runs n times for each iteration of outer loop.

hence, $$T(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} O(1)$$

$$T(n) = O(n^2)$$

② Code and image provided.

③ Upper bound : $T(n) = O(n^2)$

Lower bound : $T(n) = \Omega(n^2)$

Tight bound : $T(n) = \Theta(n^2)$

④ From the plot attached, it is shown that $n_0 \simeq 20$.

After $n_0 \simeq 20$, the runtime data starts to consistently follow the polynomial trend.

⑤ The modified function has an extra statement `y = i+j;` inside the inner loop. It will increase the number of iterations from n^2 to $2n^2$ (approx.). The time complexity will still be $O(n^2)$.

⑥ No, it will not effect the result from #1.

The time complexity will still remain the same.

⑦ Code provided.