1. O(n + 10)
   1. O(n)
2. O(100 \* n)
   1. O(n)
3. O(25)
   1. O(1)
4. O(n^2 + n^3)
   1. O(n^3)
5. O(n + n + n + n)
   1. O(n)
6. O(1000 \* log(n) + n)
   1. O(n)
7. O(1000 \* n \* log(n) + n)
   1. O(n log n)
8. O(2^n + n^2)
   1. O(2^n)
9. O(5 + 3 + 1)
   1. O(1)
10. O(n + n^(1/2) + n^2 + n \* log(n)^10)
    1. O(n^2)

Step 2

1. O(n)
2. O(n)
3. O(1)
4. O(n)
5. O(n^2)
6. O(n)

Part 3

1. True or false: n^2 + n is O(n^2).
   1. true
2. True or false: n^2 \* n is O(n^3).
   1. true
3. True or false: n^2 + n is O(n).
   1. false
4. What’s the time complexity of the .indexOf array method?
   1. O(n)
5. What’s the time complexity of the .includes array method?
   1. O(n)
6. What’s the time complexity of the .forEach array method?
   1. O(n)
7. What’s the time complexity of the .sort array method?
   1. O(n log n)
8. What’s the time complexity of the .unshift array method?
   1. O(n)
9. What’s the time complexity of the .push array method?
   1. O(1)
10. What’s the time complexity of the .splice array method?
    1. O(n)
11. What’s the time complexity of the .pop array method?
    1. O(1)
12. What’s the time complexity of the Object.keys() function?
    1. O(n)
13. What’s the space complexity of the Object.keys() function? O(n)
    1. O(n)