**Part 1**

1.

O(n+10)O(n + 10)O(n+10) → O(n)O(n)O(n)

* Drop the constant.

2.

O(100∗n)O(100 \* n)O(100∗n) → O(n)O(n)O(n)

* Drop the constant multiplier.

3.

O(25)O(25)O(25) → O(1)O(1)O(1)

* Constants simplify to O(1)O(1)O(1).

4.

O(n2+n3)O(n^2 + n^3)O(n2+n3) → O(n3)O(n^3)O(n3)

* Keep the dominant term n3n^3n3.

5.

O(n+n+n+n)O(n + n + n + n)O(n+n+n+n) → O(n)O(n)O(n)

* Combine terms: 4n4n4n simplifies to O(n)O(n)O(n).

6.

O(1000∗log⁡(n)+n)O(1000 \* \log(n) + n)O(1000∗log(n)+n) → O(n)O(n)O(n)

* nnn dominates log⁡(n)\log(n)log(n).

7.

O(1000∗n∗log⁡(n)+n)O(1000 \* n \* \log(n) + n)O(1000∗n∗log(n)+n) → O(nlog⁡(n))O(n \log(n))O(nlog(n))

* nlog⁡(n)n \log(n)nlog(n) dominates nnn.

8.

O(2n+n2)O(2^n + n^2)O(2n+n2) → O(2n)O(2^n)O(2n)

* 2n2^n2n grows faster than n2n^2n2.

9.

O(5+3+1)O(5 + 3 + 1)O(5+3+1) → O(1)O(1)O(1)

* Constants simplify to O(1)O(1)O(1).

10.

O(n+n1/2+n2+n∗log⁡(n)10)O(n + n^{1/2} + n^2 + n \* \log(n)^{10})O(n+n1/2+n2+n∗log(n)10) → O(n2)O(n^2)O(n2)

* n2n^2n2 dominates all other terms.

**Part 3**

1. **True**: n2+nn^2 + nn2+n is O(n2)O(n^2)O(n2).
2. **True**: n2∗nn^2 \* nn2∗n is O(n3)O(n^3)O(n3).
3. **False**: n2+nn^2 + nn2+n is O(n2)O(n^2)O(n2), not O(n)O(n)O(n).
4. **Time complexity of .indexOf**: O(n)O(n)O(n).
5. **Time complexity of .includes**: O(n)O(n)O(n).
6. **Time complexity of .forEach**: O(n)O(n)O(n).
7. **Time complexity of .sort**: O(nlog⁡(n))O(n \log(n))O(nlog(n)).
8. **Time complexity of .unshift**: O(n)O(n)O(n).
9. **Time complexity of .push**: O(1)O(1)O(1).
10. **Time complexity of .splice**: O(n)O(n)O(n).
11. **Time complexity of .pop**: O(1)O(1)O(1).
12. **Time complexity of Object.keys()**: O(n)O(n)O(n), where nnn is the number of keys in the object