Зайцев К0709-22/3

1. Решено на паре
2. Решено на паре
3. O(n) - первый цикл имеет сложность O(n), а второй цикл O(1), ведь он выполняется только единожды раз из-за комманды break.
4. O(n * log n) - главный цикл имеет сложность O(n/2) а вложенный в него имеет сложность O(log n), а вложенный вложенный (не знаю, как его еще назвать) цикл тоже имеет сложность O(log n), потому что итератор удваивается на каждой итерации, пока не достигнет n. в итоге получаем O(n/2 * $\log^2$n), а после отбрасывания констант получаем ответ.
5. O(n^2 * log n) - главный цикл обладает сложностью O(n/2), упростив логическое выражение во вложенном цикле можно понять, что он имеет ту же сложность, что и главный, а вложенный вложенный цикл обладает сложностью O(log n) поскольку итератор увеличивается вдвое на каждой итерации пока не достигнет n. В итоге получаем O(n/2 * n/2 * log n), далее отбрасываем константы и получаем ответ.
6. O(sqrt(n)) – путем перебора значений переменной n, можно выявить закономерность, что количество итераций квадратично зависит от увеличения переменной n (не знаю как это иначе объяснить, только перебирать значения n и приводить примеры).
7.
8. O(sqrt n) – смотреть пункт 6
9. O(n) – так, как единственный цикл в функции выполняется n раз, а остальные части кода имеют константную сложность.
10. в лучшем случае цикл не выполниться ни разу, а в худшем случае может быть вплоть до бесконечности итераций, в зависимости от того во сколько раз наибольшее число больше наименьшего.
11. O(sqrt(n)) — так как программа будет выполняться до тех пор пока i^2 меньше n.
12. O(log n)
13. O(n^2)
14. O(n^2) - главный цикл сделает n итераций, а вложенный n/i итераций. А общая сложность алгоритма будет n^2
15. O(n^2) - главный цикл имеет сложность n/3, а вложенный из за шага в 4 имеет сложность n/4, так что сложность данной программы будет O(n/3*n/4), после отбрасывания констант получаем ответ.
16. O(log n) - главный и вложенный цикл имеют сложность логарифм по основанию 2 от n, поскольку циклы вложенные мы перемножаем их сложность и получаем O(log^2 n), а после отбрасываем степень.
17. O(log n) – потому что j будет увеличиваться в 2 раза при каждой итерации, пока она не превысит n.
18. O(N + M) - так как первый цикл имеет сложность O(N), а второй имеет сложность O(M), операции внутри циклов имеют константную сложность и поэтому мы просто складываем их сложность и получаем ответ.
19. O(n^2) – количество итераций главного цикла равно n, а количество внутреннего цикла зависит от значения итератора главного цикла(а именно N-i раз). Общее количество итераций можно определить по формуле суммы первых n членов арифметической прогрессии ((N * (N+1)) / 2
20. O(n * log n) - так как главный цикл имеет сложность O(n/2), а вложенный в него O(log n). перемножаем их, отбрасываем константы и получаем ответ.
21. log(n) – при переборе значений N, можно выявить закономерность- если если значение N увеличивается в 2 раза, то количество итераций цикла увеличится на 1.
22. O(n) – так как цикл выполняется n раз, несмотря на умножение итератора на k

23. $O(n^2)$ - поскольку начальное значение итератора вложенного цикла зависит от значения итератора внешнего цикла, то сложность алгоритма можно определить по формуле суммы первых n членов арифметической прогрессии, но после отбрасывания констант получаем n^2