Introduction exercises on algorithm analysis and time complexity

Students start this assignment in groups of 4-6.

Students finish as many iterations of this exercise as they can in the time allotted.
First iterations, pick something you believe is easy, just to get started.

Each iteration is:

a) Each individual in a group picks at random (or chooses) one of the following time complexities:
O(1), **O($\log_2 n$)**, **O(n)**, **O(n*$\log_2 n$)**, **O(n^2)**, **O(n^3)**, **O(2^n)**

Each student writes down on a piece of paper a set of instructions (pseudocode is fine) for an algorithm that runs in the time complexity they chose. The algorithm does not have to correctly run a specific operation. It can be as silly as needed :)
n *can be the length of a list or the value of a variable. Students should state what* **n** *is.*

b) All individuals in the group show their result to the group. The group as a whole either confirms the time complexity or argues why it is different that the individual intended. Preferably all students in the group should agree on all the pseudoprograms for this iteration.

The following times are not sacred but try to work fast to get more varied training and exercise. Simply finish as much as you can in about 30 minutes.

The times for the iterations is as follows:
Iterations 1-3
a) 1 minute
b) 3 minutes (for whole group)

Iterations 4-5
a) 3 minutes
b) 4 minutes

For the later iterations pick some of the time complexities that are more difficult or combined etc. Students can even decide on other time complexities for added challenge.