Qurstion 1:

1. by the Big-Oh definition, T(n) is O(n^2) if T(n) ≤ c·n^2for some n ≥ n0. Let us check this condition: if n^2 + 20n + 1 ≤ c·n^2 then 1+20/n+1/n^2≤c. Therefore, the Big-Oh condition holds for n ≥ n0 = 1 and c ≥ 22.
2. By the Big-Oh definition, T(n) is O(n) if T(n) ≤ c·n for some n ≥ n0. Let us check this condition: if n^2 + 20n + 1 ≤ c·n then n+20/n≤c. Therefore, the Big-Oh condition cannot hold.

Question 2:

1. O(n^2)
2. O(n^2)
3. O(nlogn)

Question 3:

1. We first throw the egg at the N/2th floor and see if it is broken. If is, then we throw it again at the N/4th floor to see if it is broken. If not broken, we will test the 3N/4thfloor to see if it is broken and we repeat this action until we find the exact number of the floor.
2. start at 1, next 2, 4, 8 (i.e., 2^i), once the egg breaks do binary search in the smaller search space

Question 4:

1.4N+1

2.4 √𝑁+2

Question 5:

1. algorithm 1：3N^2+4N+2

algorithm 2：6N+3

algorithm 3：2N+1

1. By the picture, we can see that when the size of number increases, the runtime needs for method 1 increase the most, 3 increase second most, and method 2 becomes the most time efficient one.