1.

Create a new array a

Create a new array output

Loop through the array B{

Loop through the array a{

If the current value of B is found in a, add it to the output array and move on

}

If the current value of B isn’t found in a, add it to a

}

Return output

//This might be done better with hash, I wonder?

2.

Method that takes in int x, int y {

If y == 0{

Return 0

}else{

Return x + method(x, y-1)

}

}

3.

See attached

4.

n3, n2, 4nlogn+2n, nlogn, 3n+100logn, 4n, 2n, 2logn, 210

5.

Method (array, count, target number){

If count > length of the array, return -1

Loop through the array{

Check if the count and the current value of the array sum to the target number, return them if they do

}

Otherwise, return the method(array, count++, target number)

}

}

As indicated by the loop inside something that recurs a maximum of n times where n is the length of the array, it has a Big O of n2

6.

Give each of log n taste testers their own bottle to drink from. Next, give a bottle to each possible pair of two taste testers. Then give one bottle to each possible group of three taste testers. Repeat this until you run out of bottles or you have 1 group with all of your taste testers in it. Finally, set one bottle off to the side. If one taste tester dies, the poison is in the bottle that only they drank. If two die, it is in the bottle they shared. This continues up to the point of having \*all\* of the king’s taste testers die, meaning it is in the bottle they all shared (hence why I assume the question mentioned the poison being very deadly, to scale with large values of n the poison would be very diluted from splitting 1 bottle with many taste testers) If no taste testers have died, the poison is in the bottle set off to the side.