CoinGecko Interview Questions – Joshua Ahimaz

1. *myself.json is attached*
2. Even though an ArrayList would be the best solution to a dynamic array, it isn't possible to create a somewhat dynamic array, in my instance I have created an array that can increase the size of the array, although this is quite inefficient as compared to an ArrayList. I have created a small demonstration in Java on how this is possible with inline documentation.

   Main function ----------

```java
public static void main(String[] args) {
    //The initial array with a fixed size of 3
    int[] array = {1, 2, 3};
    //The for loop can be ignored, as it is just for demonstration purposes.
    for(int i = 4; i <= 20; i++){
        //The AddToArray function is called, two arguments are passed
        array = AddToArray(array, i);
        //The first argument is the array itself, the second is the value you would like to
        //add/append to the end of the array.
        System.out.println(Arrays.toString(array));
    }
}
```

   AddToArray function ----------

```java
//A static function that can be called at anytime.
//The AddToArray function takes has parameters, if it were its own
//Java file, it could have overloading methods to handle different
//possible values depending on the Array type
public static int[] AddToArray(int[] array, int element){
    //First a new copy of the array (with a size incremented by one)
    //is made and the previous values are placed within
    array = Arrays.copyOf(array, newLength: array.length + 1);
    //The new element value is added to the newly created array
    array[array.length - 1] = element;
    //The array is returned replacing the older array
    return array;
}
```

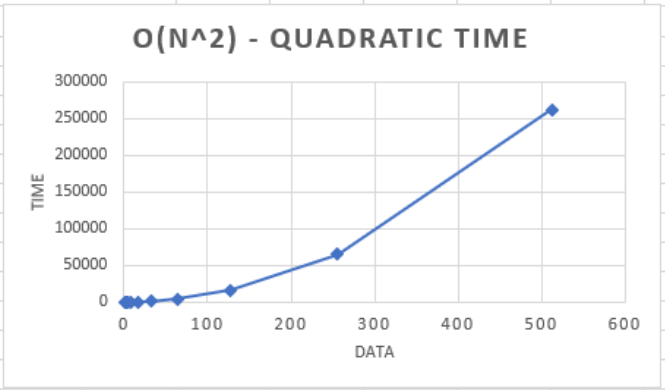
Example output, adding 17 new values to the array.

```
[1, 2, 3]
[1, 2, 3, 4]
[1, 2, 3, 4, 5]
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 4, 5, 6, 7]
[1, 2, 3, 4, 5, 6, 7, 8]
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
```

3. This function represents the $O(n^2)$ algorithm (Also known as Quadratic time), which involves nesting two loops. For every *n* number of items in the array, the outer loop of the array will run *n* times, and the inner loop will run *n* times for each iteration of the outer loop, hence a n^2 number of output (or print in this scenario). A programming scenario would be a bubble sort.

Made by Joshua Ahimaz

Data	Time
1	1
2	4
4	16
8	64
16	256
32	1024
64	4096
128	16384
256	65536
512	262144



Some examples with values:

- Array with 20 items, print 400 times
- Array with 512 items, print 262144 times

4. A closure is both the enclosed function together with its references to its surrounding state or better known as the **lexical environment**, closures give you access to the outer function's scope within the inner function. For JS these closures are created every time a function itself is created at function creation time. In order to use a closure, you will have to define the function inside another function and expose it, to expose this function, return it or pass it on to another function. The inner functions will have access to the variables in the outer function scope, even if it has returned.

   Examples in which we use closures:

   Data Privacy: Data privacy is important to program for an interface rather than an implementation. When used for privacy, you can't get data from an outside scope except through the methods.

   self-invoking function: The self-invoking function will run once, set its counter to 0 and return the expression.

5.

	var	let	const
Scope	Global if defined outside a block, locally scoped if defined within a function.	Scoped within a block ({ }), defining outside a block will throw an error.	Scoped within a block ({ }), defining outside a block will throw an error.
Hoisting (variables and function declarations move to the top during code execution)	**var** gets hoisted during code execution. Initialised with *undefined*	**let** gets hoisted but doesn't get initialized, leading to a reference error if used before declaration.	**const** gets hoisted but doesn't get initialized, leading to a reference error if used before declaration.

Update or Re-declaration	Possible to re-declare and update within the scape scope.	Cannot be re-declared but can be updated.	Cannot be re-declared or updated.
Example Usages	Usage in a global scope, to be accessed throughout the code. Although if a repetition of the var identifier occurs in large codes, this could lead to bugs.	Good for temporary variables within a function. That have no reason to be accessed outside of the function/block scope.	Const is best used when you have a defined constant that has no reason to change later on.

6. The book that had changed my perspective on life is "*the pursuit of happyness*". One of my favorite reads as it shows the hardships that Chris Gardner had to face as a single father to survive every single day, and push forward to become a stock broker, even as hard as of a path it was with no education. His only dedication was happiness. The book has taught me not to take things for granted and that life can push you down many, many times even till rock bottom, but you will still have a chance to climb your way back up.
7. I believe proper technological advancement is priority to everything else such as poverty. Everybody wants to live the normal life of having a house, a car, a computer, but people do not realise that this could have such a negative impact on the world this could have, in terms of pollution caused by everyone having a car. Of course this does not mean there have to be people who will suffer or live in poverty forever. The point is that we need to focus on technological advancement to solve global problems, not solve global problems while advancing in technology.
8. As someone who had studied both arts and humanities in my high school. art is expression done by individuals, whereas humanities is focused on the individuals. I think arts and humanities is and always will play an important role in society. We

cannot escape art; it is so formless that anything can be art and it is such a wonderful thing. A painting, movie, dance, and a revolution are all art. Humanities is the discipline of human values and our way to express (not an expression such as art but rather the act of expressing), this is equally important as it is how we live by everyday of our present life, the philosophy and how we understood our past, the history.