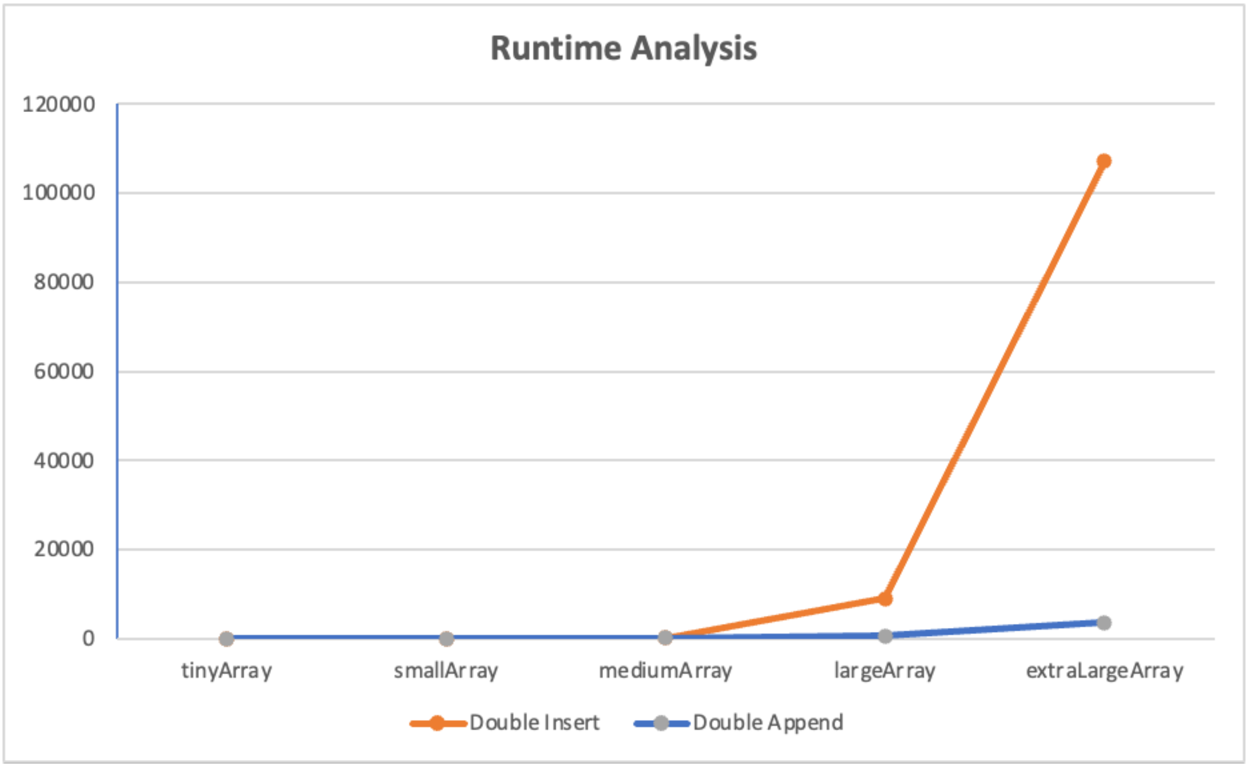
Array	Size of Array	Double Insert	Double Append
`tinyArray`	10	24	65
`smallArray`	100	34.1	89.3
`mediumArray`	1000	207.4	128.6
`largeArray`	10000	8988.2	588.04
`extraLargeArra`	100000	107099	3636.79



After collecting data of all the arrays, I converted all of them in microseconds so I can get them graphed correctly. After seeing the graph it looks like, doubleAppend scales perfectly with all the arrays but doubleInsert scales better with smaller arrays(10-1000). The **Array.push()** has a Constant Time Complexity and so is O(1). It adds an element and give it an index that's 1 greater than the index of the last element in the array. So it doesn't matter whether the size of the array is 10 or 100000. The number of operations that needs to be performed won't change. And **Array.unshift()** has a Linear Time Complexity and is O(n). Adding an element at the beginning of an array means the new element will have an index of 0. Which means that the index of every other element must be incremented by 1. So doubleAppend works much better that doubleInsert because Array.unshift takes a lot time compare to Array.push().