|  |  |
| --- | --- |
| **Array List** | **Vector** |
| Array lists are not synchronized meaning multiple threads can access the list at a given time | Vectors are synchronized such that only one thread can access the vector at a given time |
| An array list grows 50% of its current size in case the number of elements exceed capacity | A vector grows 100% of its current size in case the number of elements exceed capacity |
| Array lists are faster as they are not synchronized | Vectors are generally slower as they synchronized |
| Array lists are traversed via Iterators only | Vector can be traversed via Enumeration and Iterators |

|  |  |
| --- | --- |
| **HashSet** | **Sorted Set** |
| Unordered collection of unique elements | Ordered collection of unique elements |
| Lookups are faster | It has to do linear search for lookups which is slower |

|  |  |
| --- | --- |
| **TreeSet** | **HashSet** |
| Elements are maintained in sorted order | Elements are not sorted |
| TreeSet does not allow null objects | HashSet allows null objects |
| Uses compareTo() method for comparing object | Uses equals() method to compare objects |

|  |  |
| --- | --- |
| **Array** | **List** |
| Array needs a continuous chunk of memory allocated | A list does not need a continuous memory space. Since we use nodes with pointers to the next node, the memory can be in different places. |
| Random access | Sequential access |
| Static memory allocation | Dynamic memory allocation |

|  |  |
| --- | --- |
| **List** | **Set** |
| List allows duplicates | Set does not allow duplicates |
| Lists allow null values | Sets do not allow nulls |

|  |  |
| --- | --- |
| **Navigable Set** | **Navigable Map** |
| Stores distinct elements | Stores key value pairs |