

1.

Eine ArrayList speichert die Elemente in einem Array. Den Vorteil den ArrayLists bieten ist, dass auf spezielle Elemente schnell über die Position zugegriffen werden kann. Die Zeit bleibt dabei konstant egal wie groß das Array ist. Elemente können nur in der Mitte eingefügt werden, wenn alle nachfolgenden Elemente verschoben werden. Sie sollten bei Daten genutzt werden, die mehr geschrieben als gelesen werden.

LinkedList hingegen speichert die Elemente in einer verketteten Liste. Der Nachteil von LinkedLists ist, dass das Durchsuchen der Liste sehr zeitaufwendig ist. Dafür kann man jedoch einfach Elemente löschen oder einfügen. Sie sollten bei Daten genutzt werden, die mehr gelesen als geschrieben werden.

2.

Auf den Graphen zeigt sich kein klarer Gewinner. Jede Collection hat Gebiete auf denen sie gut und schwach sind. Das einzige was deutlich ist, ist dass CopyOnWriteArrayList in vielen Gebieten schwach ist.

3.

Erklären lässt sich das dadurch, dass bei jeder Veränderung eine Kopie erzeugt wird. Dadurch werden unnötig viele Daten verarbeitet, die nicht verändert werden.

4.

Java bietet mehrere thread safe collections. Eine davon wäre CopyOnWriteArrayList.

5.

Der Iterator benutzt eine eigene remove Methode und nicht die remove Methode der Collection.

6.

Die remove Methode der Collection entfernt die Referenz aus der Collection.

7.

In diesem Beispiel wird der Prozess nur auf einem Core ausgeführt. Die restlichen 7 bleiben unbenutzt. Um die volle Leistung des PC's zu nutzen sollte die Collection in 8 Teile aufgeteilt werden.