Performance Analyse

20.000 Werte einfügen Vone (Jedes mal der Integerwert 1)		
Durchlauf	ArrayList	DoubleLinkedList
1	1157	689
2	890	603
3	844	641
4	782	578
5	703	547
6	734	516
7	750	578
8	766	656
9	781	1188
10	782	594
Avarage Time	818	659

20.000 Werte einfügen Mitte (Jedes mal der Integerwert 1)		
Durchlauf	ArrayList	DoubleLinkedList
1	719	1203
2	703	1297
3	781	1266
4	750	1250
5	844	1234
6	750	1157
7	844	1141
8	773	1047
9	750	1172
10	782	1156
Avarage Time	769	1192

20.000 Werte einfügen Hinten (Jedes mal der Integerwert 1)		
Durchlauf	ArrayList	DoubleLinkedList
1	922	938
2	817	906
3	953	859
4	828	844
5	860	844
6	984	857
7	730	875
8	766	891
9	875	953
10	860	938
Avarage Time	859	890

1.000 Werte löschen Vorne (Jedes mal der Integerwert 1)		
Durchlauf	ArrayList	DoubleLinkedList
1	78	93
2	63	94
3	47	94
4	78	110
5	62	109
6	62	109
7	63	94
8	63	109
9	78	94
10	79	109
Avarage Time	67	101

1.000 Werte löschen Mitte (Jedes mal der Integerwert 1)		
Durchlauf	ArrayList	DoubleLinkedList
1	63	94
2	79	94
3	62	93
4	62	94
5	94	109
6	62	94
7	63	94
8	63	93
9	94	94
10	62	141
Avarage Time	70	100

1.000 Werte löschen Hinten (Jedes mal der Integerwert 1)		
Durchlauf	ArrayList	DoubleLinkedList
1	62	108
2	78	141
3	78	122
4	62	107
5	78	110
6	78	109
7	63	109
8	62	94
9	78	109
10	78	109
Avarage Time	71	111

MinMax-Algorithmus

Min = der minimale Wert

Max = der maximale Wert

Temp = nächster Wert

1. Min = Erster Wert und Max = Erster Wert
2. Temp = nächster Wert
3. Vergleiche Temp mit Min
    a. Wenn Min > Temp, dann Min = Temp. Und gehe zu 5.
    b. Sonst gehe zu 4.
4. Vergleiche Temp mit Max
    a. Wenn Max < Temp, dann Max = Temp
5. Wenn noch ein weiteres Element in der Liste ist, dann gehe zu 2, sonst breche ab.

Maximale Laufzeit = 2n