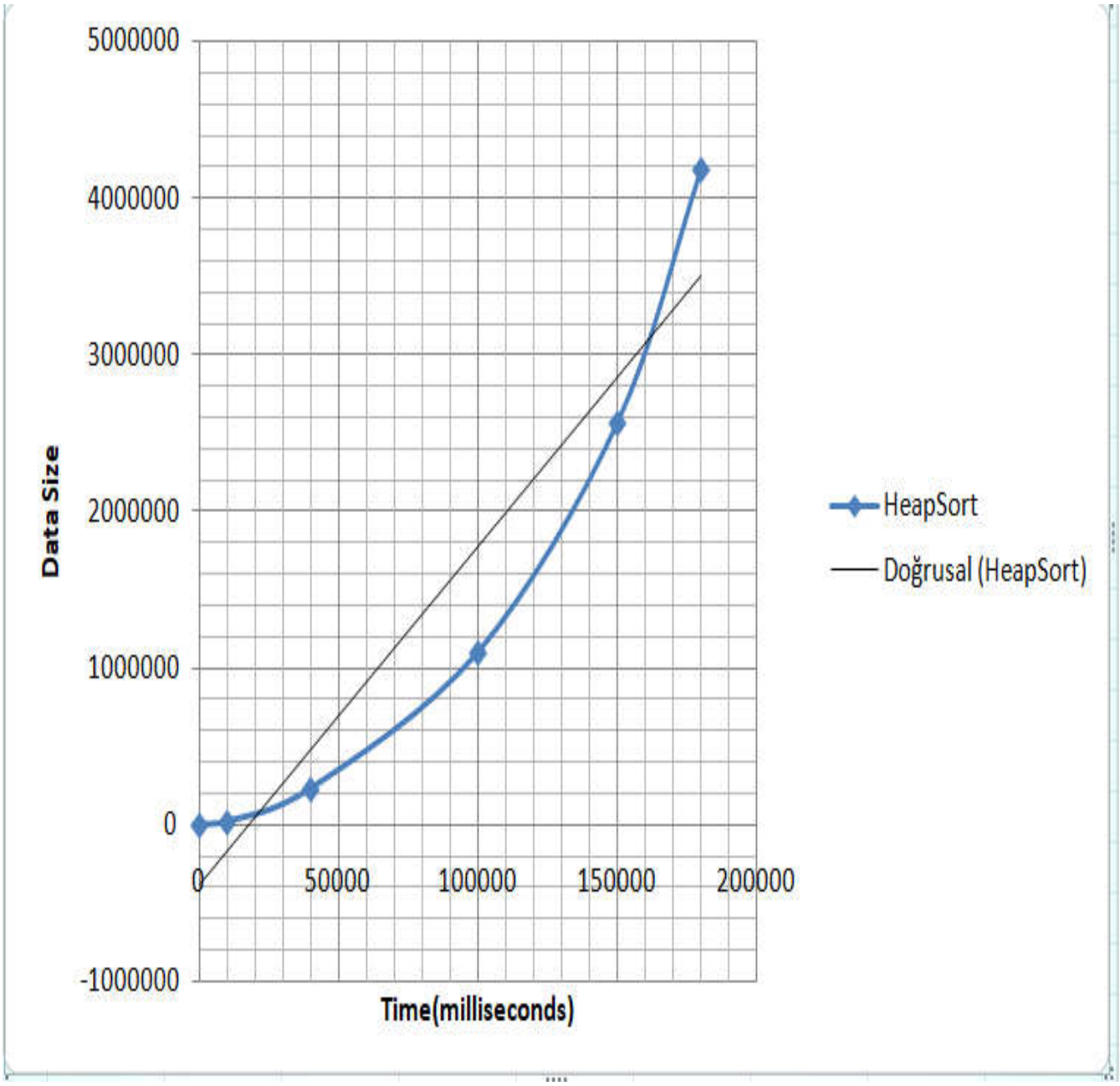
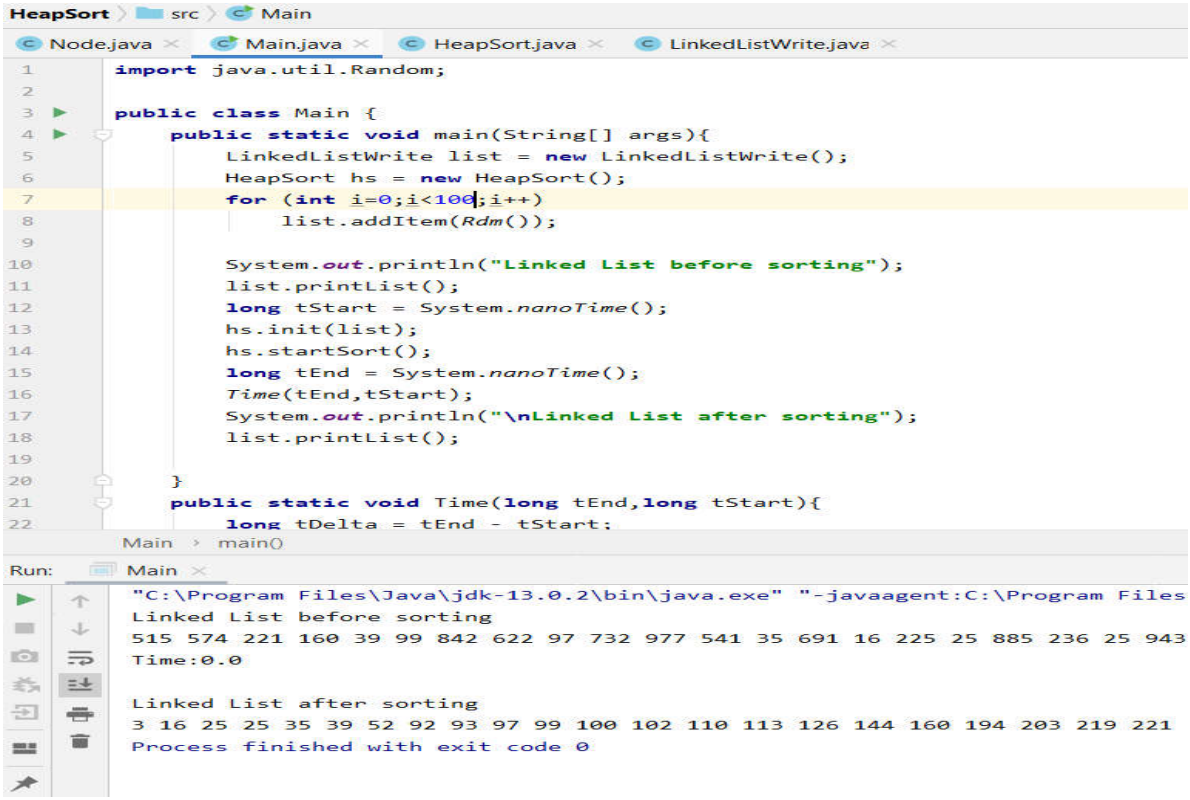
HeapSort

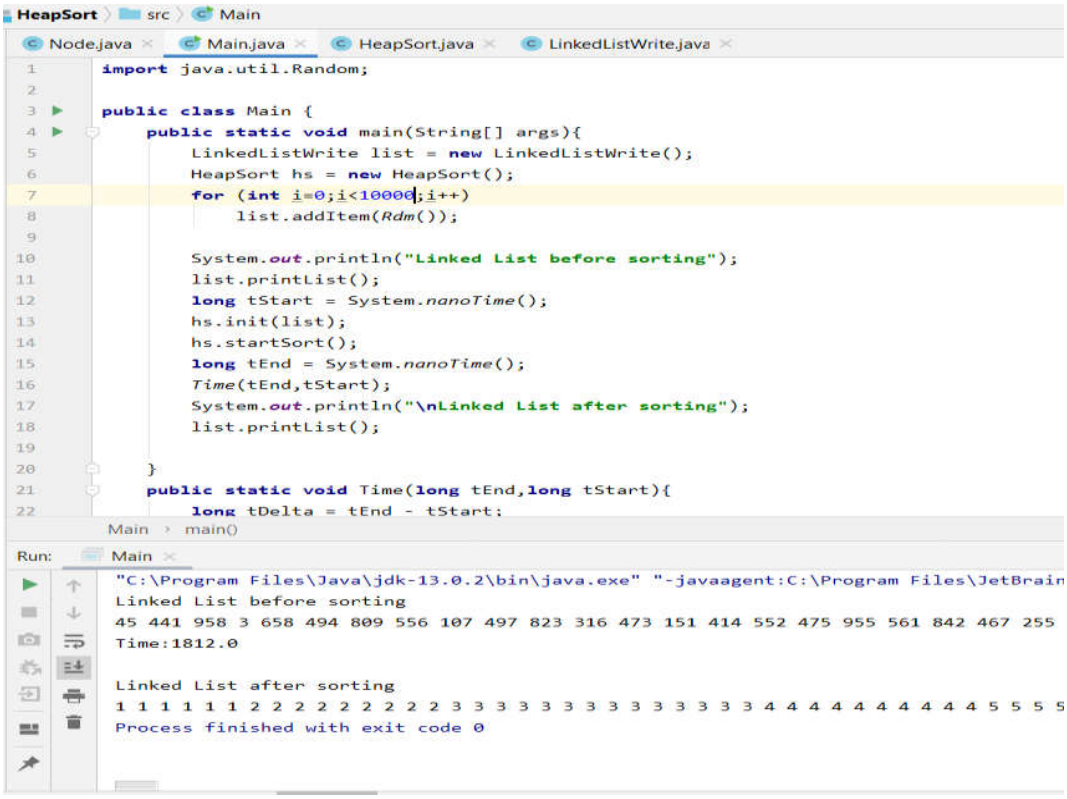

5000000
4000000
3000000
2000000
1000000
0
-1000000
Data Size
0
50000
100000
150000
200000
Time(milliseconds)
HeapSort
Doğrusal (HeapSort)

A)Heapsort-100



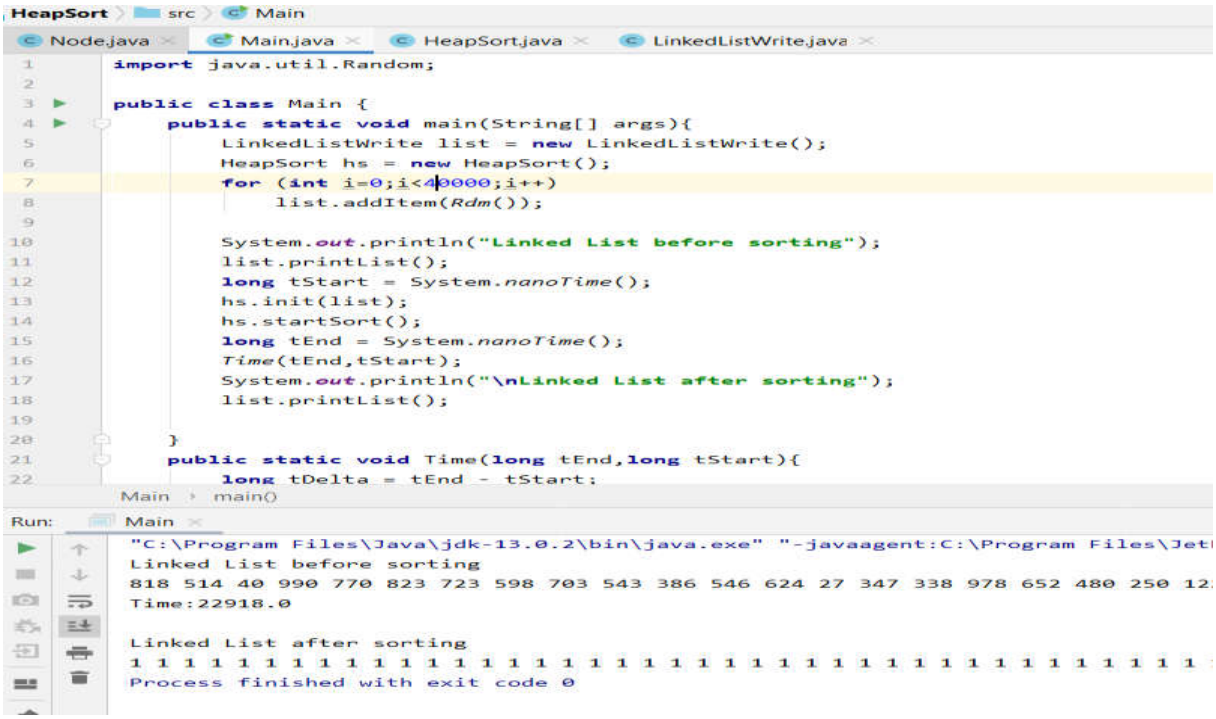
```
HeapSort > src > Main
Node.java | Main.java | HeapSort.java | LinkedListWrite.java

1  import java.util.Random;
2
3  public class Main {
4      public static void main(String[] args){
5          LinkedListWrite list = new LinkedListWrite();
6          HeapSort hs = new HeapSort();
7          for (int i=0;i<100;i++)
8              list.addItem(Rdm());
9
10         System.out.println("Linked List before sorting");
11         list.printList();
12         long tStart = System.nanoTime();
13         hs.init(list);
14         hs.startSort();
15         long tEnd = System.nanoTime();
16         Time(tEnd,tStart);
17         System.out.println("\nLinked List after sorting");
18         list.printList();
19
20     }
21     public static void Time(long tEnd,long tStart){
22         long tDelta = tEnd - tStart;

Main > main()

Run: Main
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files
Linked List before sorting
515 574 221 160 39 99 842 622 97 732 977 541 35 691 16 225 25 885 236 25 943
Time:0.0

Linked List after sorting
3 16 25 25 35 39 52 92 93 97 99 100 102 110 113 126 144 160 194 203 219 221
Process finished with exit code 0
```

B) Heapsort-10000



```
HeapSort > src > Main
Node.java | Main.java | HeapSort.java | LinkedListWrite.java

1  import java.util.Random;
2
3  public class Main {
4      public static void main(String[] args){
5          LinkedListWrite list = new LinkedListWrite();
6          HeapSort hs = new HeapSort();
7          for (int i=0;i<10000;i++)
8              list.addItem(Rdm());
9
10         System.out.println("Linked List before sorting");
11         list.printList();
12         long tStart = System.nanoTime();
13         hs.init(list);
14         hs.startSort();
15         long tEnd = System.nanoTime();
16         Time(tEnd,tStart);
17         System.out.println("\nLinked List after sorting");
18         list.printList();
19
20     }
21     public static void Time(long tEnd,long tStart){
22         long tDelta = tEnd - tStart;

Main > main()

Run: Main
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrain
Linked List before sorting
45 441 958 3 658 494 809 556 107 497 823 316 473 151 414 552 475 955 561 842 467 255
Time:1812.0

Linked List after sorting
1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5
Process finished with exit code 0
```

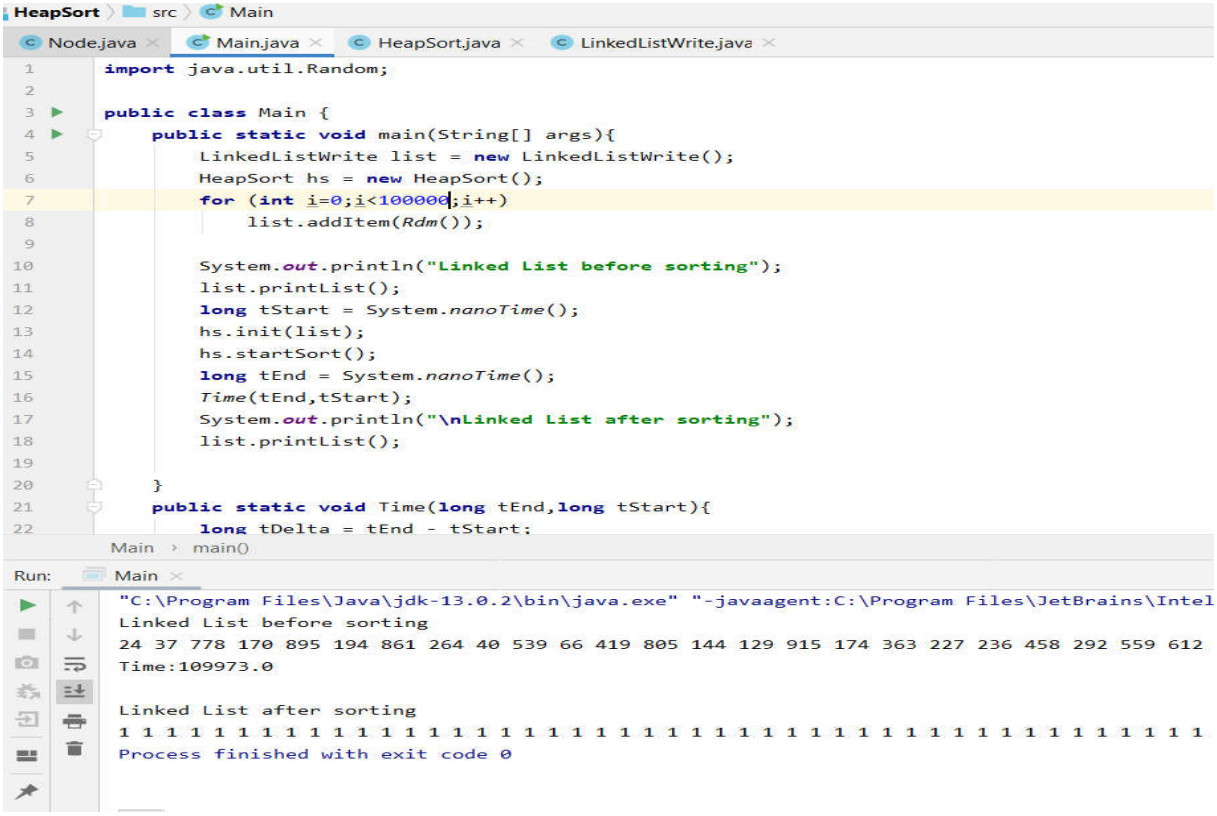
C) Heapsort-40000



```
import java.util.Random;

public class Main {
    public static void main(String[] args){
        LinkedListWrite list = new LinkedListWrite();
        HeapSort hs = new HeapSort();
        for (int i=0;i<40000;i++)
            list.addItem(Rdm());

        System.out.println("Linked List before sorting");
        list.printList();
        long tStart = System.nanoTime();
        hs.init(list);
        hs.startSort();
        long tEnd = System.nanoTime();
        Time(tEnd,tStart);
        System.out.println("\nLinked List after sorting");
        list.printList();

    }
    public static void Time(long tEnd,long tStart){
        long tDelta = tEnd - tStart;
```

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\Jet
Linked List before sorting
818 514 40 990 770 823 723 598 703 543 386 546 624 27 347 338 978 652 480 250 12
Time:22918.0

Linked List after sorting
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Process finished with exit code 0
```

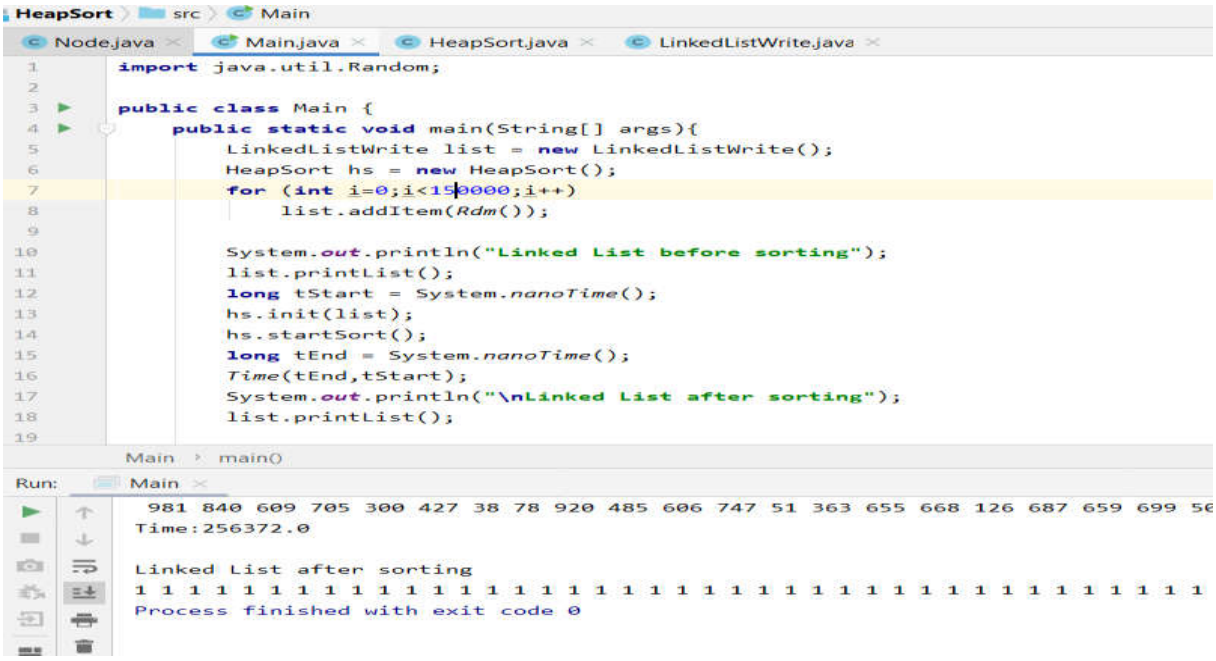
D) Heapsort-100000



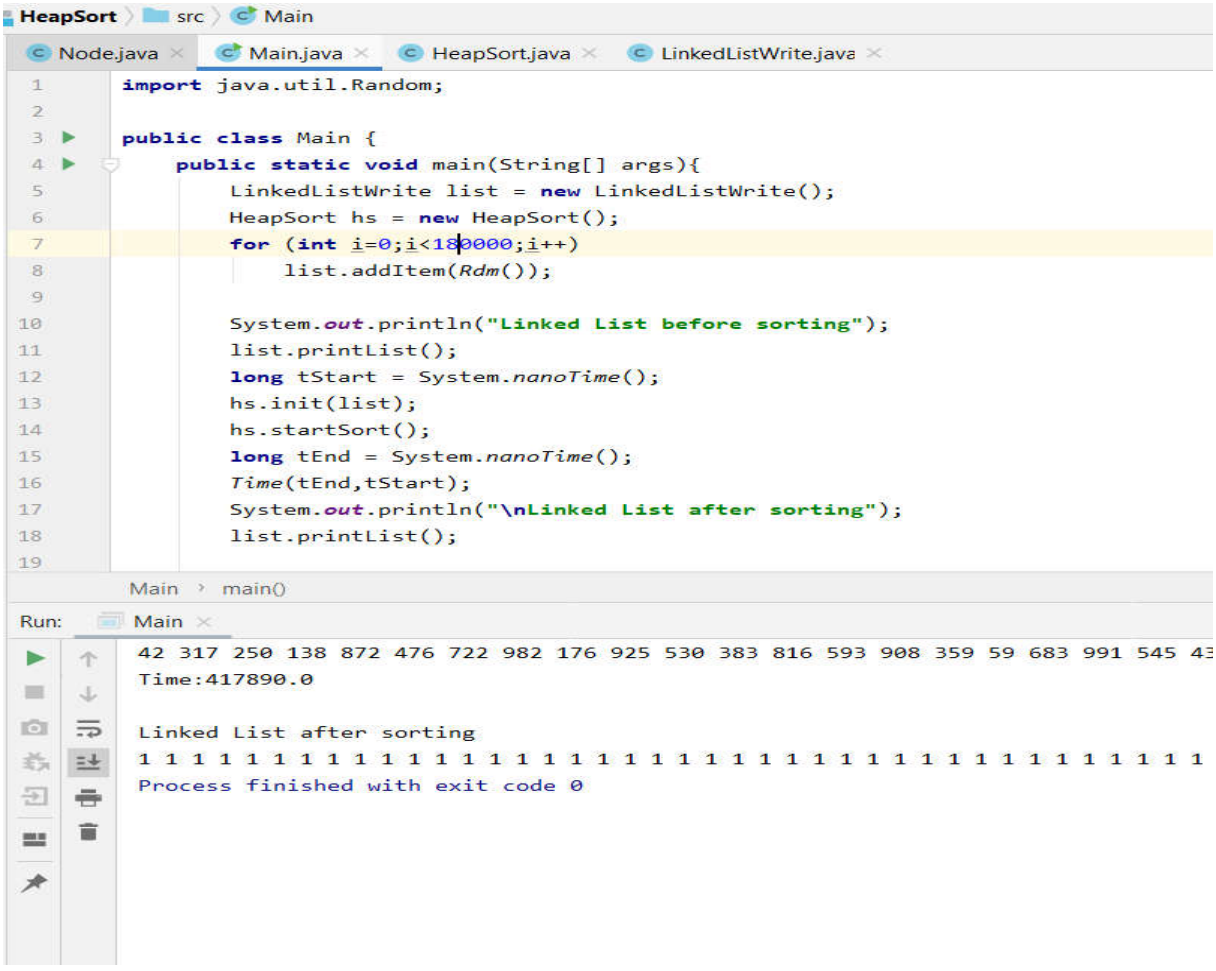
```
import java.util.Random;

public class Main {
    public static void main(String[] args){
        LinkedListWrite list = new LinkedListWrite();
        HeapSort hs = new HeapSort();
        for (int i=0;i<100000;i++)
            list.addItem(Rdm());

        System.out.println("Linked List before sorting");
        list.printList();
        long tStart = System.nanoTime();
        hs.init(list);
        hs.startSort();
        long tEnd = System.nanoTime();
        Time(tEnd,tStart);
        System.out.println("\nLinked List after sorting");
        list.printList();

    }
    public static void Time(long tEnd,long tStart){
        long tDelta = tEnd - tStart;
```

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\Intel
Linked List before sorting
24 37 778 170 895 194 861 264 40 539 66 419 805 144 129 915 174 363 227 236 458 292 559 612
Time:109973.0

Linked List after sorting
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Process finished with exit code 0
```

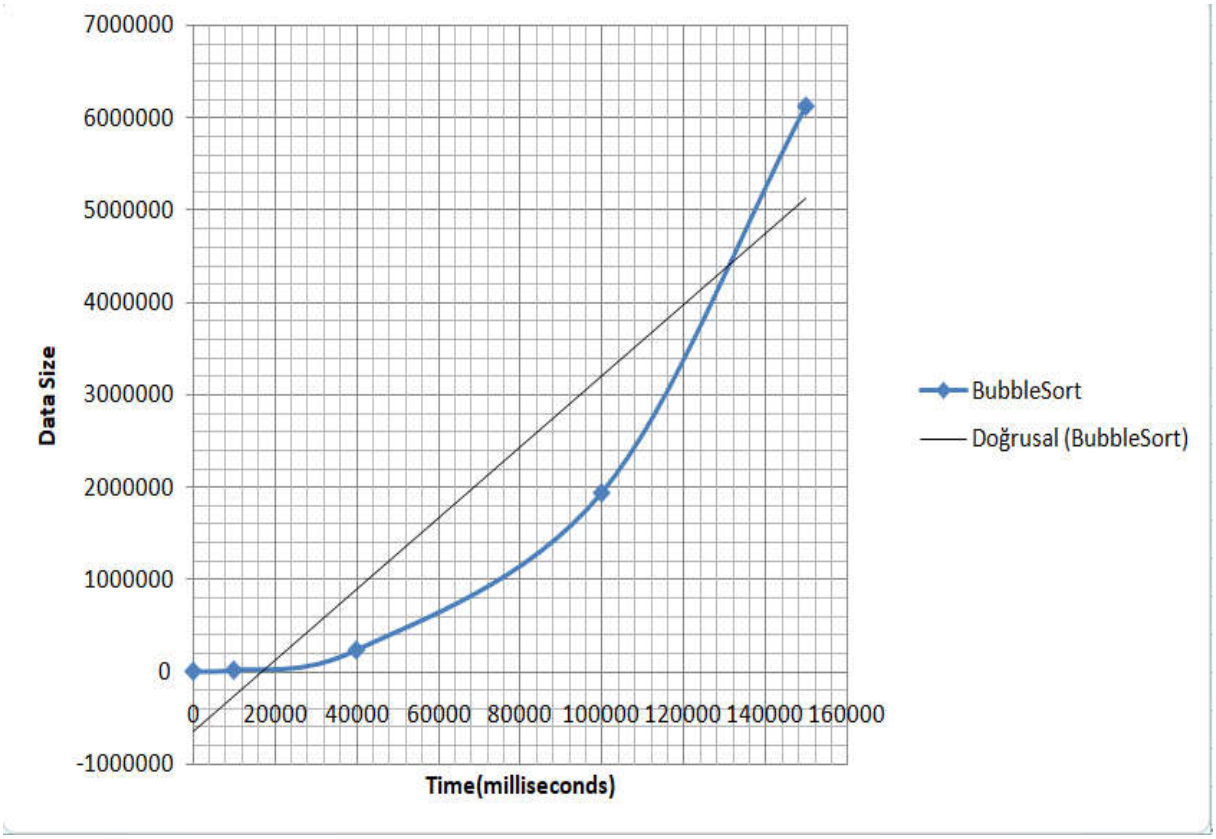
E) Heapsort-150000



F) Heapsort-180000



BubbleSort



A)BubbleSort-100

Node.java | Main.java | BubbleSort.java

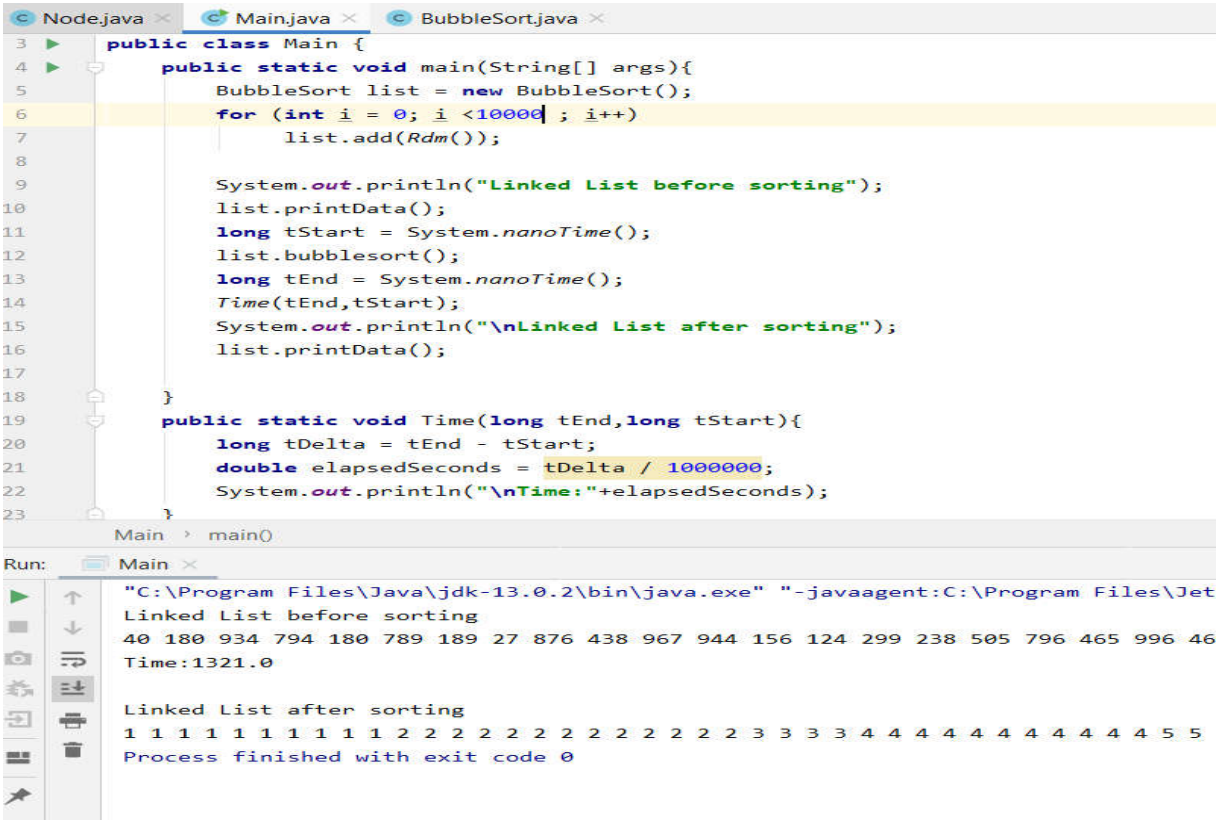
```
public class Main {
    public static void main(String[] args){
        BubbleSort list = new BubbleSort();
        for (int i = 0; i <100 ; i++)
            list.add(Rdm());

        System.out.println("Linked List before sorting");
        list.printData();
        long tStart = System.nanoTime();
        list.bubblesort();
        long tEnd = System.nanoTime();
        Time(tEnd,tStart);
        System.out.println("\nLinked List after sorting");
        list.printData();

    }
    public static void Time(long tEnd,long tStart){
        long tDelta = tEnd - tStart;
        double elapsedSeconds = tDelta / 1000000;
        System.out.println("\nTime:"+elapsedSeconds);
    }
```

Main › main()

Main

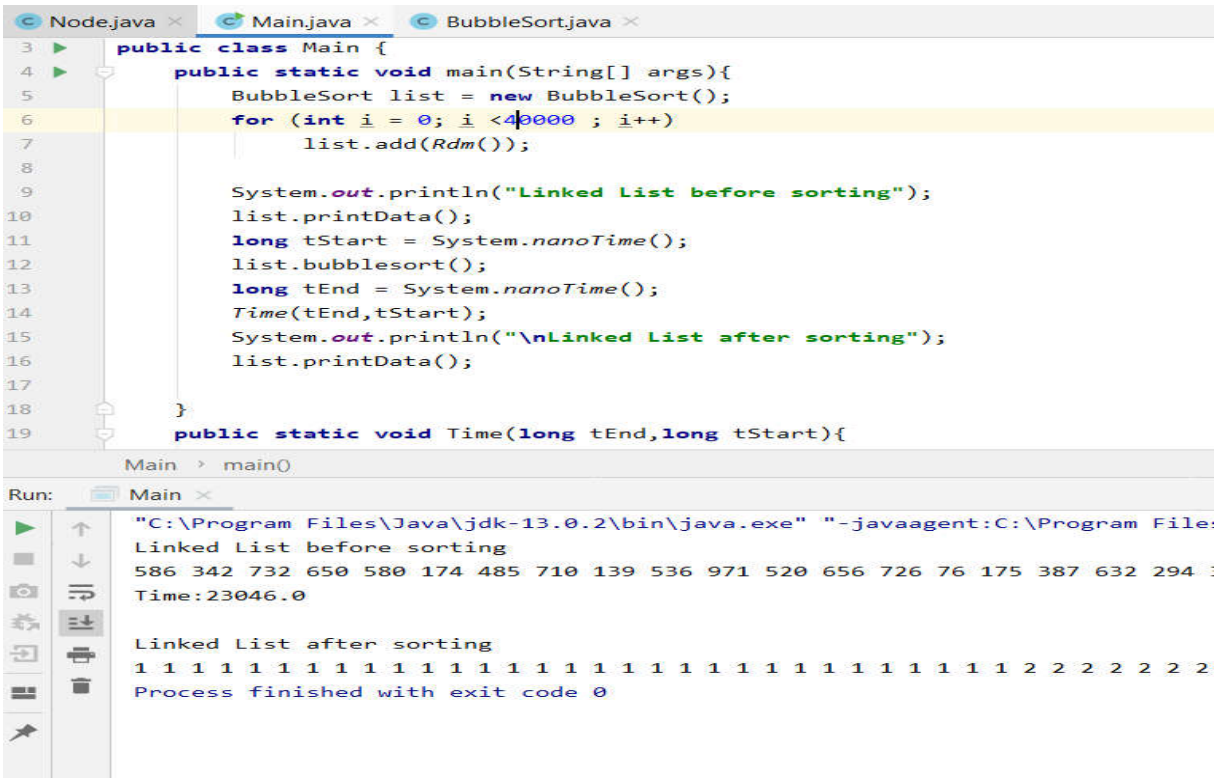
```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Prograr
Linked List before sorting
986 895 293 467 631 553 954 421 1 135 91 467 441 193 206 822 377 646
Time:0.0

Linked List after sorting
1 5 12 21 27 44 49 50 54 91 101 112 119 135 135 158 161 169 180 193 1
Process finished with exit code 0
```

B)BubbleSort-10000



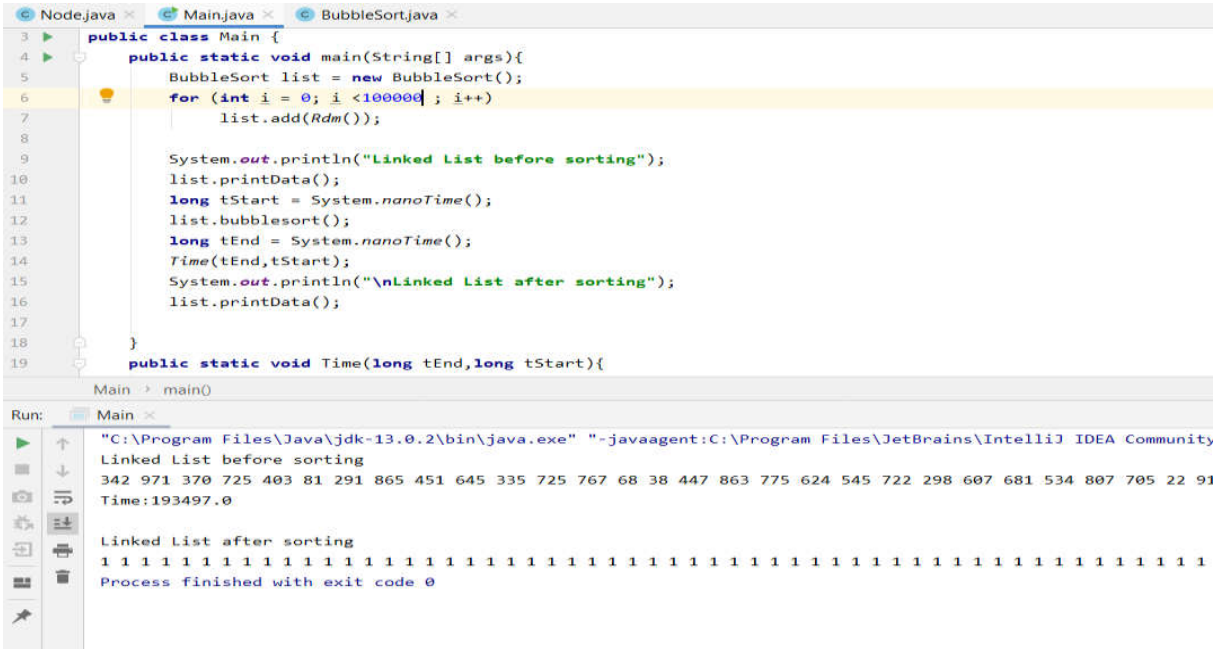
```
Node.java  Main.java  BubbleSort.java
3  public class Main {
4      public static void main(String[] args){
5          BubbleSort list = new BubbleSort();
6          for (int i = 0; i <10000 ; i++)
7              list.add(Rdm());
8
9          System.out.println("Linked List before sorting");
10         list.printData();
11         long tStart = System.nanoTime();
12         list.bubblesort();
13         long tEnd = System.nanoTime();
14         Time(tEnd,tStart);
15         System.out.println("\nLinked List after sorting");
16         list.printData();
17
18     }
19     public static void Time(long tEnd,long tStart){
20         long tDelta = tEnd - tStart;
21         double elapsedSeconds = tDelta / 1000000;
22         System.out.println("\nTime:"+elapsedSeconds);
23     }
   Main › main()
Run:  Main
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\Jet
Linked List before sorting
40 180 934 794 180 789 189 27 876 438 967 944 156 124 299 238 505 796 465 996 46
Time:1321.0

Linked List after sorting
1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 4 4 4 4 4 4 4 4 4 4 5 5
Process finished with exit code 0
```

C)BubbleSort-40000



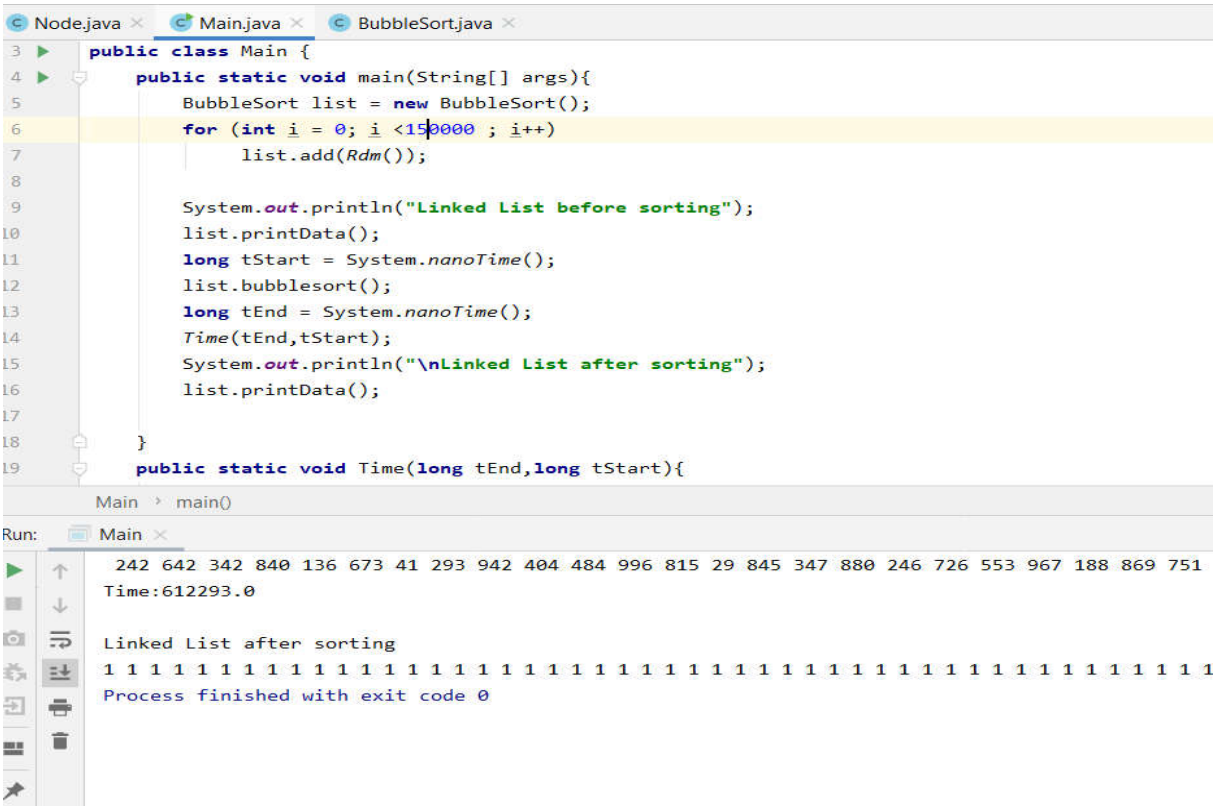
```
Node.java  Main.java  BubbleSort.java
3  public class Main {
4      public static void main(String[] args){
5          BubbleSort list = new BubbleSort();
6          for (int i = 0; i <40000 ; i++)
7              list.add(Rdm());
8
9          System.out.println("Linked List before sorting");
10         list.printData();
11         long tStart = System.nanoTime();
12         list.bubblesort();
13         long tEnd = System.nanoTime();
14         Time(tEnd,tStart);
15         System.out.println("\nLinked List after sorting");
16         list.printData();
17
18     }
19     public static void Time(long tEnd,long tStart){
   Main › main()
Run:  Main
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program File
Linked List before sorting
586 342 732 650 580 174 485 710 139 536 971 520 656 726 76 175 387 632 294
Time:23046.0

Linked List after sorting
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2
Process finished with exit code 0
```

D)BubbleSort- 100000



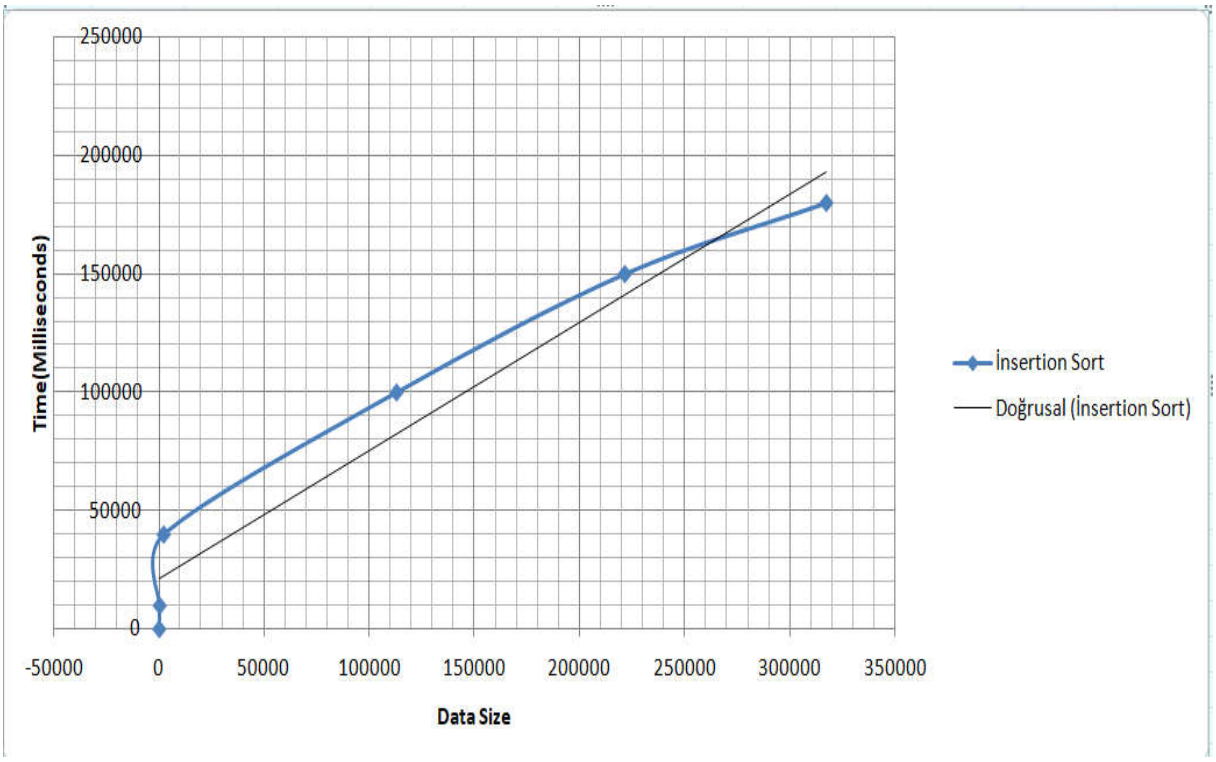
```
Node.java  Main.java  BubbleSort.java
3  public class Main {
4      public static void main(String[] args){
5          BubbleSort list = new BubbleSort();
6          for (int i = 0; i <100000 ; i++)
7              list.add(Rdm());
8
9          System.out.println("Linked List before sorting");
10         list.printData();
11         long tStart = System.nanoTime();
12         list.bubblesort();
13         long tEnd = System.nanoTime();
14         Time(tEnd,tStart);
15         System.out.println("\nLinked List after sorting");
16         list.printData();
17
18     }
19     public static void Time(long tEnd,long tStart){

Main › main()

Run:  Main
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
Linked List before sorting
342 971 370 725 403 81 291 865 451 645 335 725 767 68 38 447 863 775 624 545 722 298 607 681 534 807 705 22 91
Time:193497.0

Linked List after sorting
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Process finished with exit code 0
```

E)BubbleSort-150000



```
Node.java  Main.java  BubbleSort.java
3  public class Main {
4      public static void main(String[] args){
5          BubbleSort list = new BubbleSort();
6          for (int i = 0; i <150000 ; i++)
7              list.add(Rdm());
8
9          System.out.println("Linked List before sorting");
10         list.printData();
11         long tStart = System.nanoTime();
12         list.bubblesort();
13         long tEnd = System.nanoTime();
14         Time(tEnd,tStart);
15         System.out.println("\nLinked List after sorting");
16         list.printData();
17
18     }
19     public static void Time(long tEnd,long tStart){

Main › main()

Run:  Main
 242 642 342 840 136 673 41 293 942 404 484 996 815 29 845 347 880 246 726 553 967 188 869 751
Time:612293.0

Linked List after sorting
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Process finished with exit code 0
```

InsertionSort



A)InsertionSort-100

Node.java | insertionsortLinkedList.java | Main.java

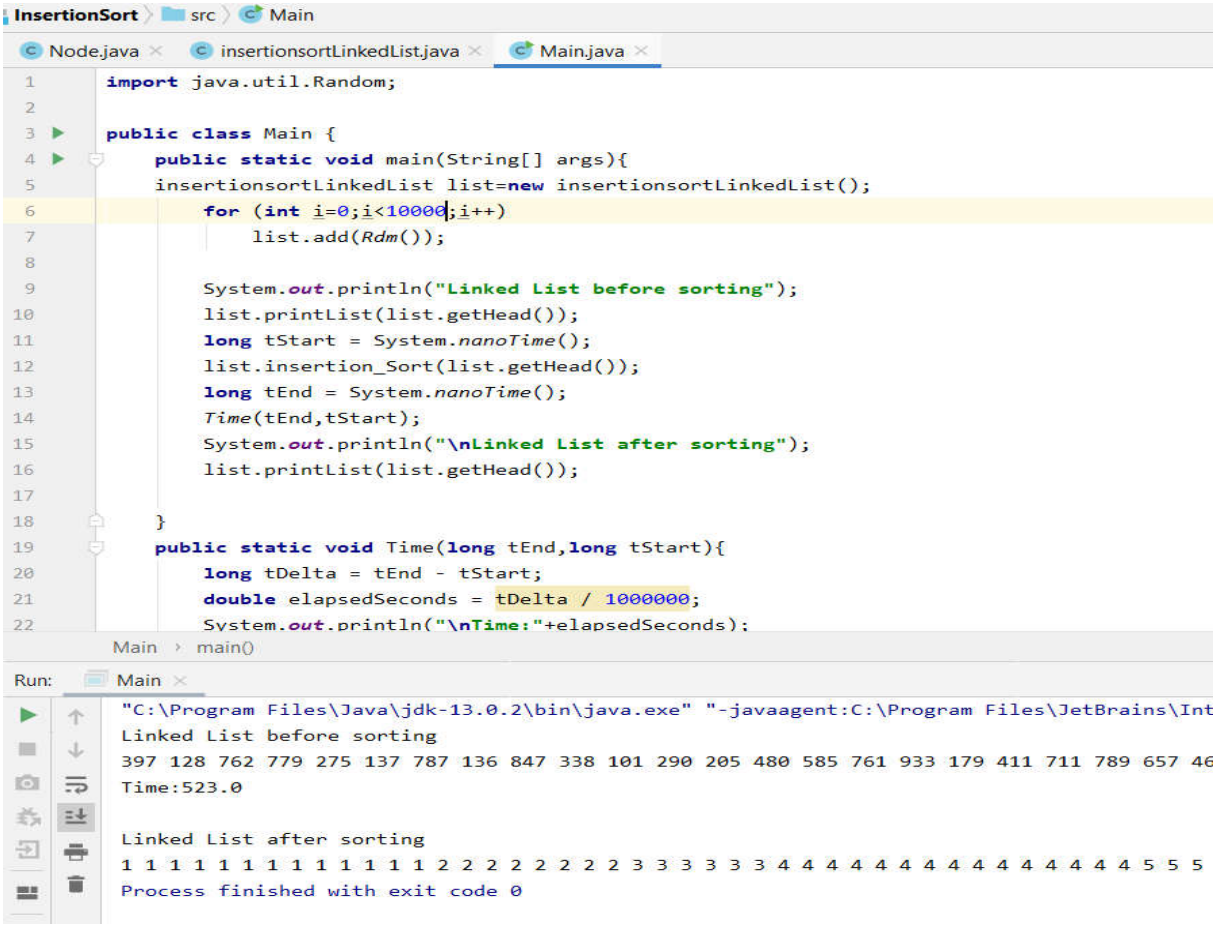
```
1   import java.util.Random;
2
3   public class Main {
4       public static void main(String[] args){
5       insertionsortLinkedList list=new insertionsortLinkedList();
6           for (int i=0;i<100;i++)
7               list.add(Rdm());
8
9           System.out.println("Linked List before sorting");
10          list.printList(list.getHead());
11          long tStart = System.nanoTime();
12          list.insertion_Sort(list.getHead());
13          long tEnd = System.nanoTime();
14          Time(tEnd,tStart);
15          System.out.println("\nLinked List after sorting");
16          list.printList(list.getHead());
17
18      }
19      public static void Time(long tEnd,long tStart){
20          long tDelta = tEnd - tStart;
21          double elapsedSeconds = tDelta / 1000000;
22          System.out.println("\nTime:"+elapsedSeconds);
```

Main › main()

Run: Main

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\Int
Linked List before sorting
905 821 851 122 417 907 830 266 549 606 44 225 326 857 61 418 4 32 438 413 746 776 313 162
Time:0.0

Linked List after sorting
1 4 7 22 26 28 32 41 44 51 61 65 71 94 96 99 120 122 135 148 158 161 162 170 175 181 185 1
Process finished with exit code 0
```

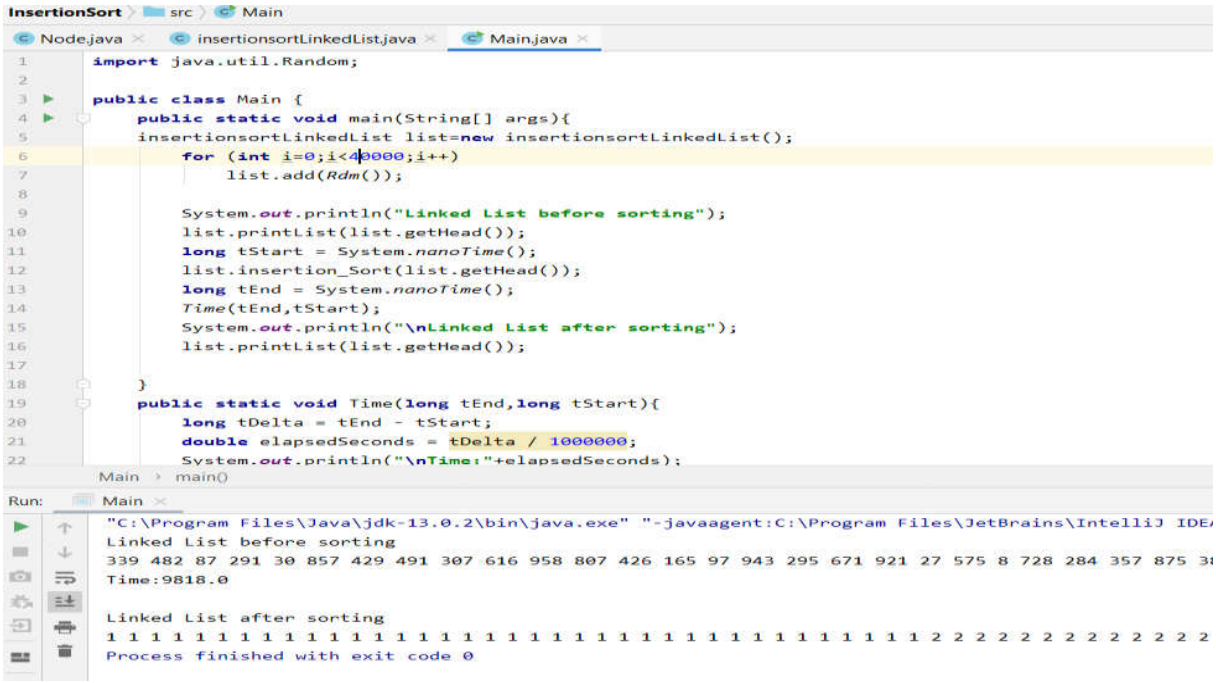
B)InsertionSort-10000



```
import java.util.Random;

public class Main {
    public static void main(String[] args){
    insertionsortLinkedList list=new insertionsortLinkedList();
        for (int i=0;i<10000;i++)
            list.add(Rdm());

        System.out.println("Linked List before sorting");
        list.printList(list.getHead());
        long tStart = System.nanoTime();
        list.insertion_Sort(list.getHead());
        long tEnd = System.nanoTime();
        Time(tEnd,tStart);
        System.out.println("\nLinked List after sorting");
        list.printList(list.getHead());

    }
    public static void Time(long tEnd,long tStart){
        long tDelta = tEnd - tStart;
        double elapsedSeconds = tDelta / 1000000;
        System.out.println("\nTime:"+elapsedSeconds);
```

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\Int
Linked List before sorting
397 128 762 779 275 137 787 136 847 338 101 290 205 480 585 761 933 179 411 711 789 657 46
Time:523.0

Linked List after sorting
1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5
Process finished with exit code 0
```

C)InsertionSort-40000



```
import java.util.Random;

public class Main {
    public static void main(String[] args){
    insertionsortLinkedList list=new insertionsortLinkedList();
        for (int i=0;i<40000;i++)
            list.add(Rdm());

        System.out.println("Linked List before sorting");
        list.printList(list.getHead());
        long tStart = System.nanoTime();
        list.insertion_Sort(list.getHead());
        long tEnd = System.nanoTime();
        Time(tEnd,tStart);
        System.out.println("\nLinked List after sorting");
        list.printList(list.getHead());

    }
    public static void Time(long tEnd,long tStart){
        long tDelta = tEnd - tStart;
        double elapsedSeconds = tDelta / 1000000;
        System.out.println("\nTime:"+elapsedSeconds);
```

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDE
Linked List before sorting
339 482 87 291 30 857 429 491 307 616 958 807 426 165 97 943 295 671 921 27 575 8 728 284 357 875 3
Time:9818.0

Linked List after sorting
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
Process finished with exit code 0
```

D)InsertionSort-100000

Node.java | insertionsortLinkedList.java | Main.java

```
import java.util.Random;

public class Main {
    public static void main(String[] args){
    insertionsortLinkedList list=new insertionsortLinkedList();
        for (int i=0;i<100000;i++)
            list.add(Rdm());

        System.out.println("Linked List before sorting");
        list.printList(list.getHead());
        long tStart = System.nanoTime();
        list.insertion_Sort(list.getHead());
        long tEnd = System.nanoTime();
        Time(tEnd,tStart);
        System.out.println("\nLinked List after sorting");
        list.printList(list.getHead());

    }
    public static void Time(long tEnd,long tStart){
        long tDelta = tEnd - tStart;
        double elapsedSeconds = tDelta / 1000000;
        System.out.println("\nTime:"+elapsedSeconds);
```

Main › main()

Run: Main

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ ID
Linked List before sorting
502 704 40 878 543 522 902 200 867 785 359 672 860 898 17 485 781 707 429 305 929 367 69 307 414 6
Time:72543.0

Linked List after sorting
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Process finished with exit code 0
```

E)InsertionSort-150000

InsertionSort › src › Main

Node.java | insertionsortLinkedList.java | Main.java

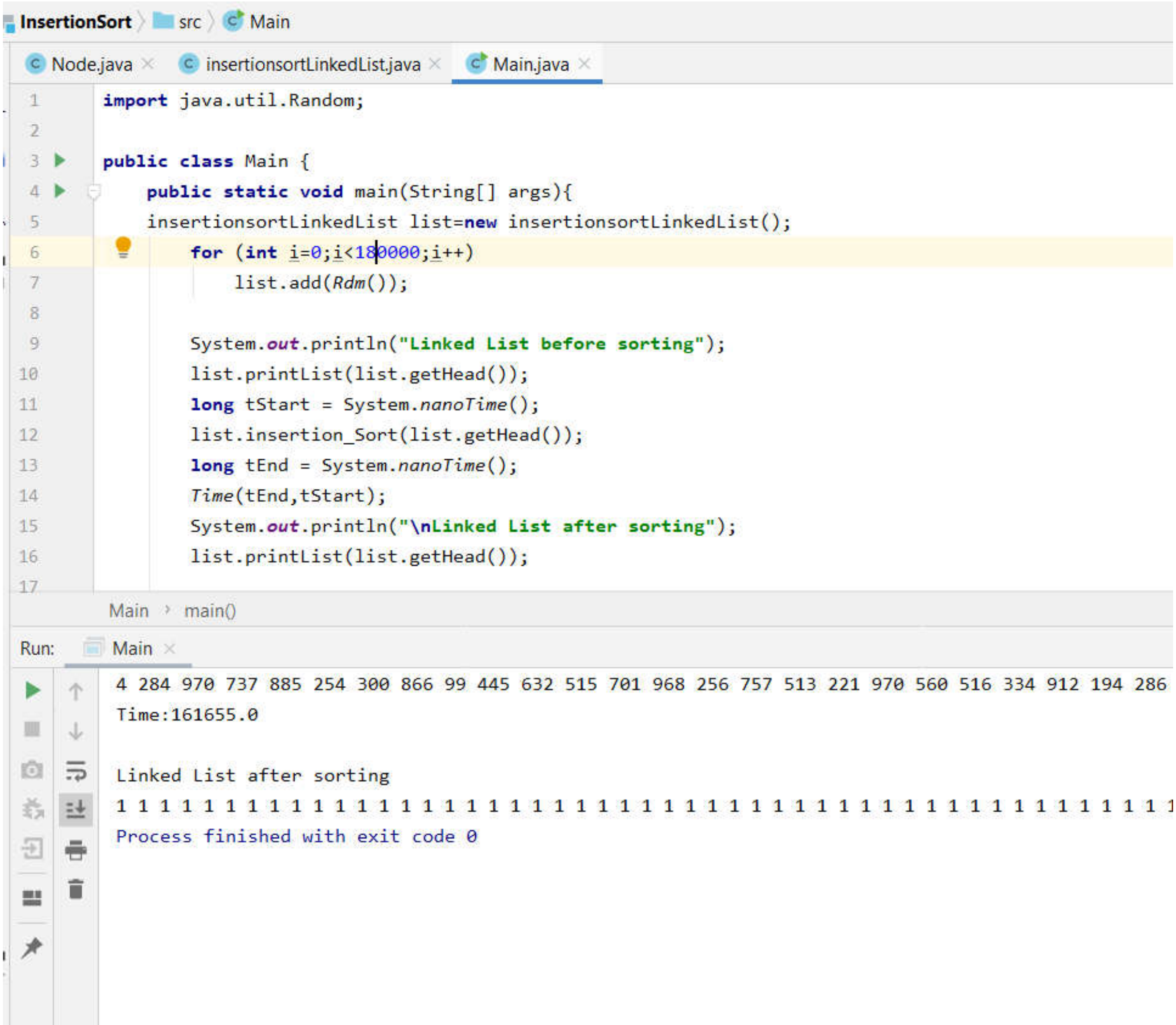
```
import java.util.Random;

public class Main {
    public static void main(String[] args){
    insertionsortLinkedList list=new insertionsortLinkedList();
        for (int i=0;i<150000;i++)
            list.add(Rdm());

        System.out.println("Linked List before sorting");
        list.printList(list.getHead());
        long tStart = System.nanoTime();
        list.insertion_Sort(list.getHead());
        long tEnd = System.nanoTime();
        Time(tEnd,tStart);
        System.out.println("\nLinked List after sorting");
        list.printList(list.getHead());
```

Main › main()

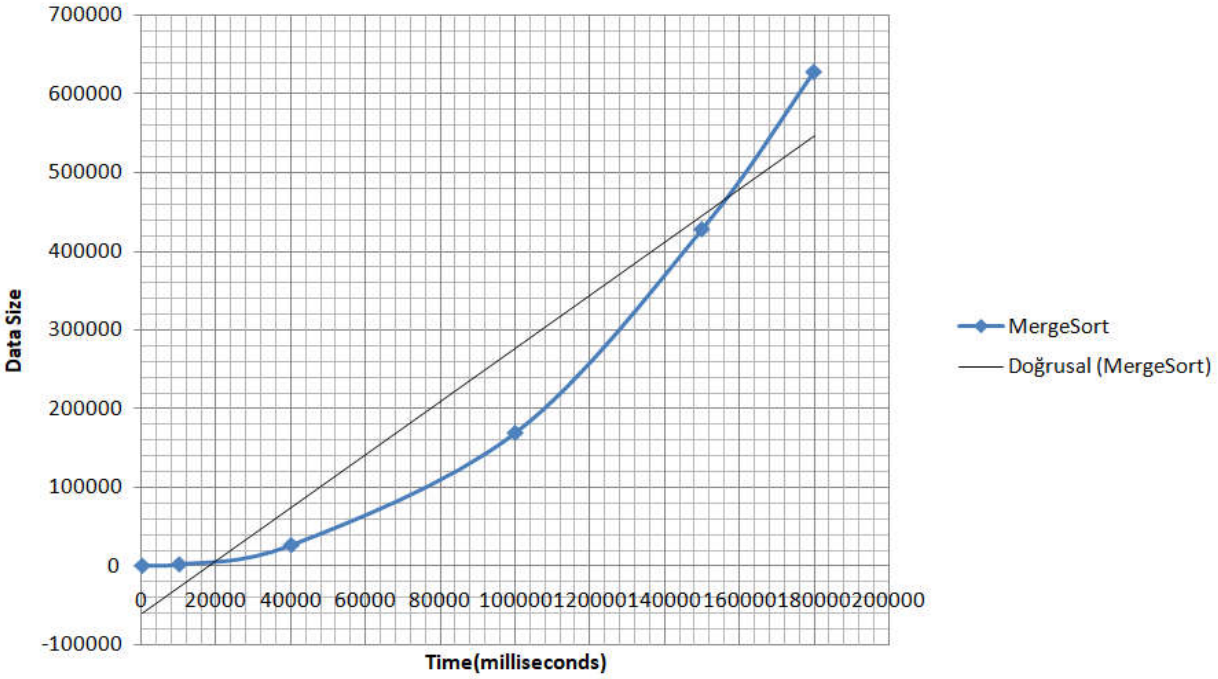
Run: Main

```
446 71 173 300 923 830 428 353 357 395 909 406 592 453 129 69 549 172 408 300 374 431 277 470 610 376 703 889
Time:247423.0

Linked List after sorting
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Process finished with exit code 0
```

F)InsertionSort-180000



MergeSort



A) MergeSort-100

MergeSort › src › Main

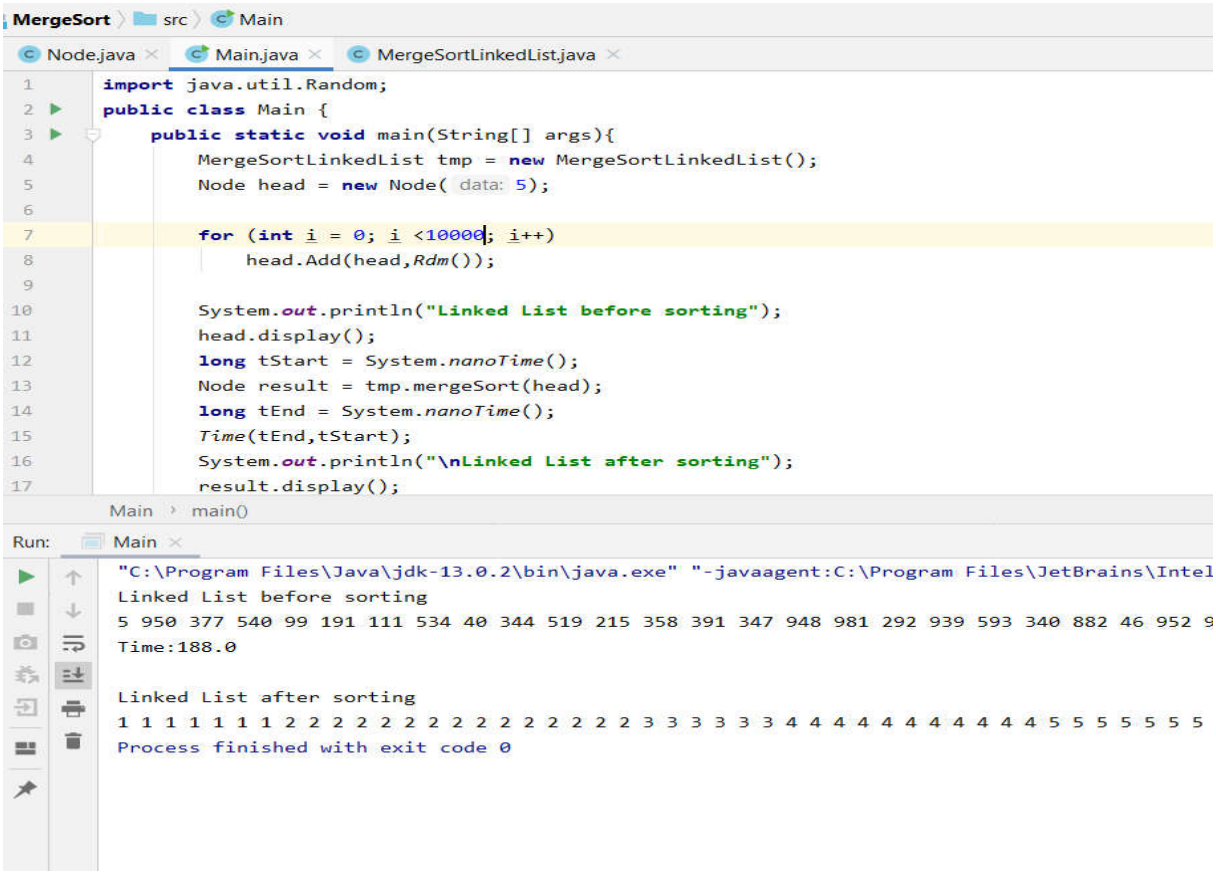
Node.java | Main.java | MergeSortLinkedList.java

```
1  import java.util.Random;
2  public class Main {
3      public static void main(String[] args){
4          MergeSortLinkedList list = new MergeSortLinkedList();
5          for (int i = 0; i <100; i++)
6              list.Add(Rdm());
7
8          System.out.println("Linked List before sorting");
9          list.printList();
10         long tStart = System.nanoTime();
11         list.setHead(list.mergeSort(list.getHead()));
12         long tEnd = System.nanoTime();
13         Time(tEnd,tStart);
14         System.out.println("\nLinked List after sorting");
15         list.printList();
16
17
18
19
20     }
21     public static void Time(long tEnd,long tStart){
22         long tDelta = tEnd - tStart;
```

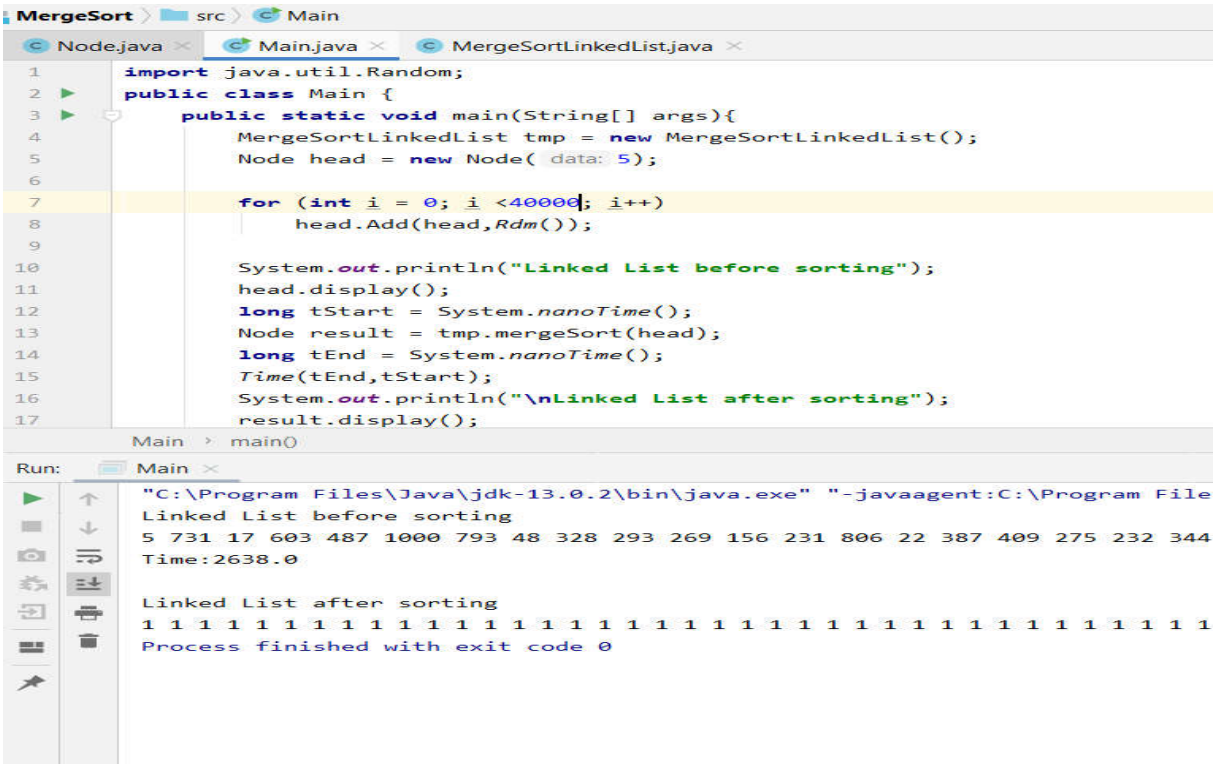
Run: Main

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2019.3.3\lib\idea_
Linked List before sorting
401 168 685 14 140 738 540 760 162 469 368 195 38 501 569 734 299 674 144 702 305 822 728 862 783 112 381 32 626 857 637 836 177 187 738 5
Time:0.0

Linked List after sorting
3 12 14 27 29 32 38 38 47 52 61 62 62 72 88 112 130 140 144 151 153 162 168 177 181 187 191 195 218 235 237 239 239 251 255 255 286 299 3
Process finished with exit code 0
```

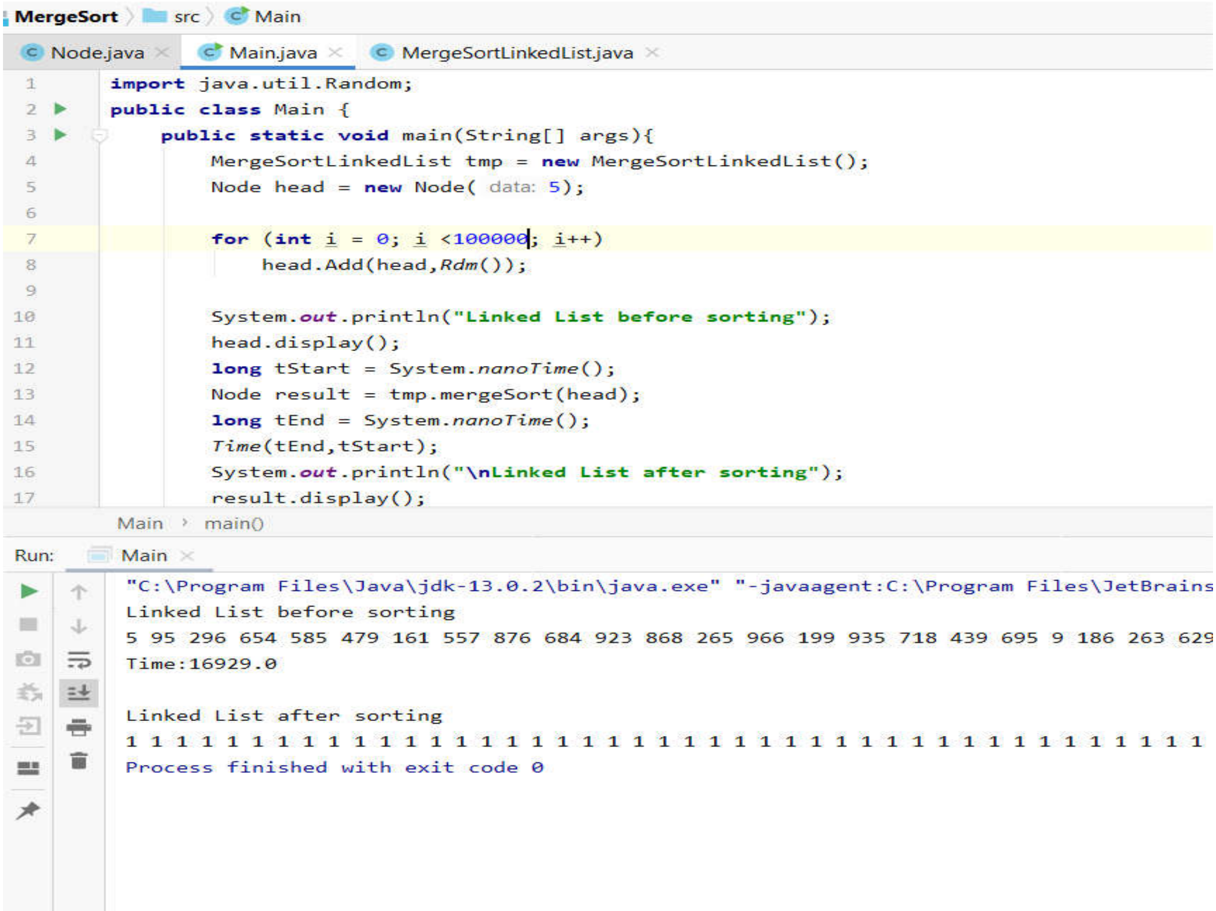
B) MergeSort-10000



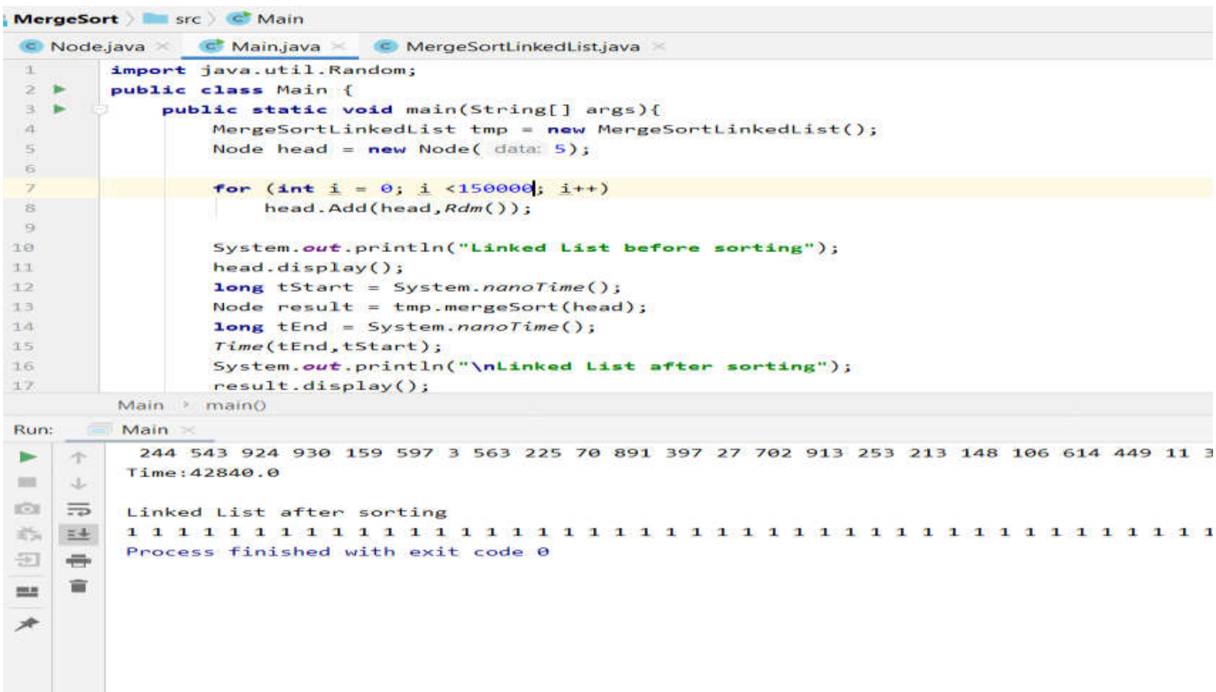
C) MergeSort-40000



D) MergeSort-100000



E) MergeSort-150000



F) MergeSort-180000

MergeSort 〉 src 〉 Main

Node.java | Main.java | MergeSortLinkedList.java

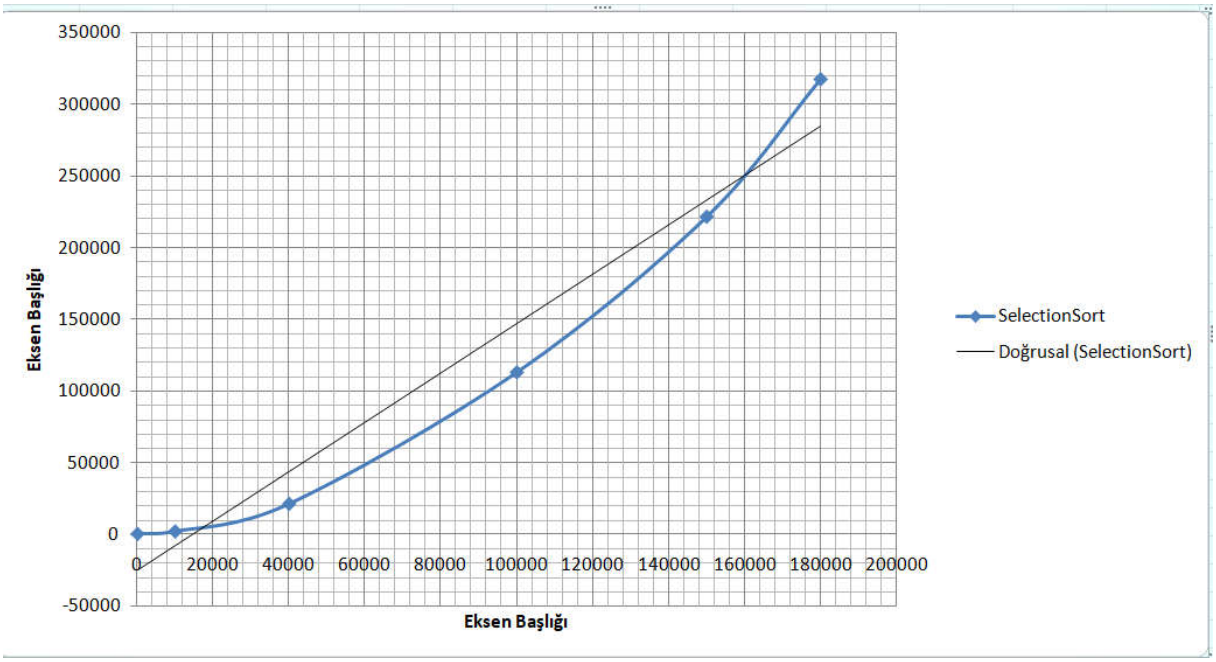
```java
import java.util.Random;
public class Main {
    public static void main(String[] args){
        MergeSortLinkedList tmp = new MergeSortLinkedList();
        Node head = new Node( data: 5);

        for (int i = 0; i <180000; i++)
            head.Add(head,Rdm());

        System.out.println("Linked List before sorting");
        head.display();
        long tStart = System.nanoTime();
        Node result = tmp.mergeSort(head);
        long tEnd = System.nanoTime();
        Time(tEnd,tStart);
        System.out.println("\nLinked List after sorting");
        result.display();
```

Main › main()

Run: Main

```
919 709 842 404 836 86 285 520 909 990 792 433 571 450 72 935 125 760 72 326 315 131 786 405 166 302 981 957
Time:62871.0

Linked List after sorting
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Process finished with exit code 0
```

SelectionSort



A) SelectionSort-100

Main.java | SelectionSortLinkList.java | Node.java

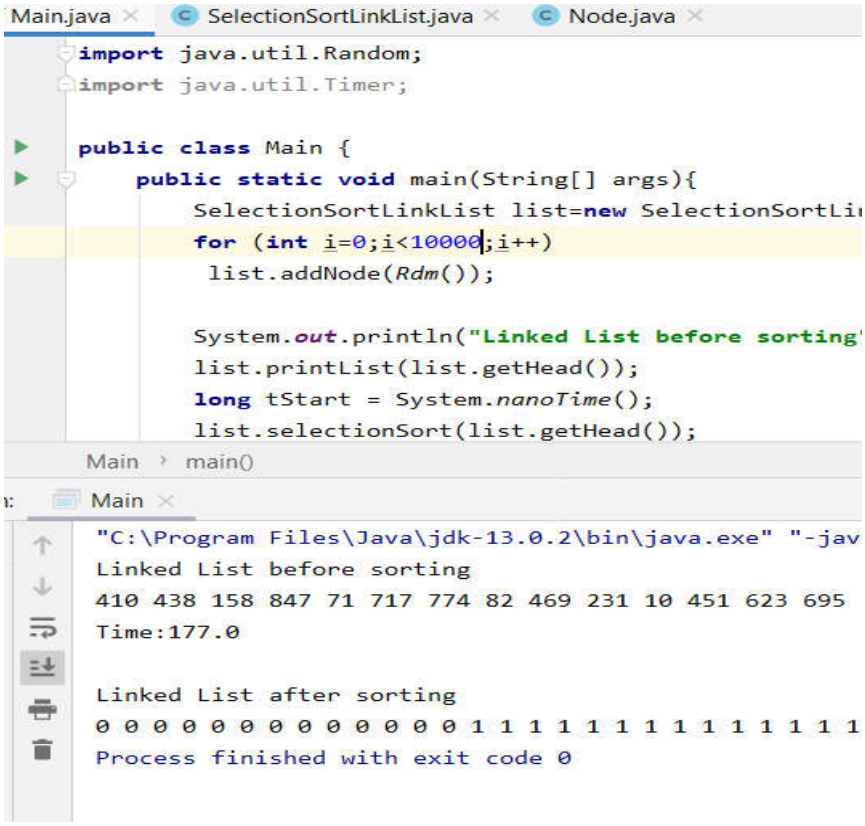
```
import java.util.Random;
import java.util.Timer;

public class Main {
    public static void main(String[] args){
        SelectionSortLinkList list=new SelectionSortLinkList();
        for (int i=0;i<100;i++)
         list.addNode(Rdm());

        System.out.println("Linked List before sorting");
        list.printList(list.getHead());
        long tStart = System.nanoTime();
        list.selectionSort(list.getHead());
```

Main › main()

Main

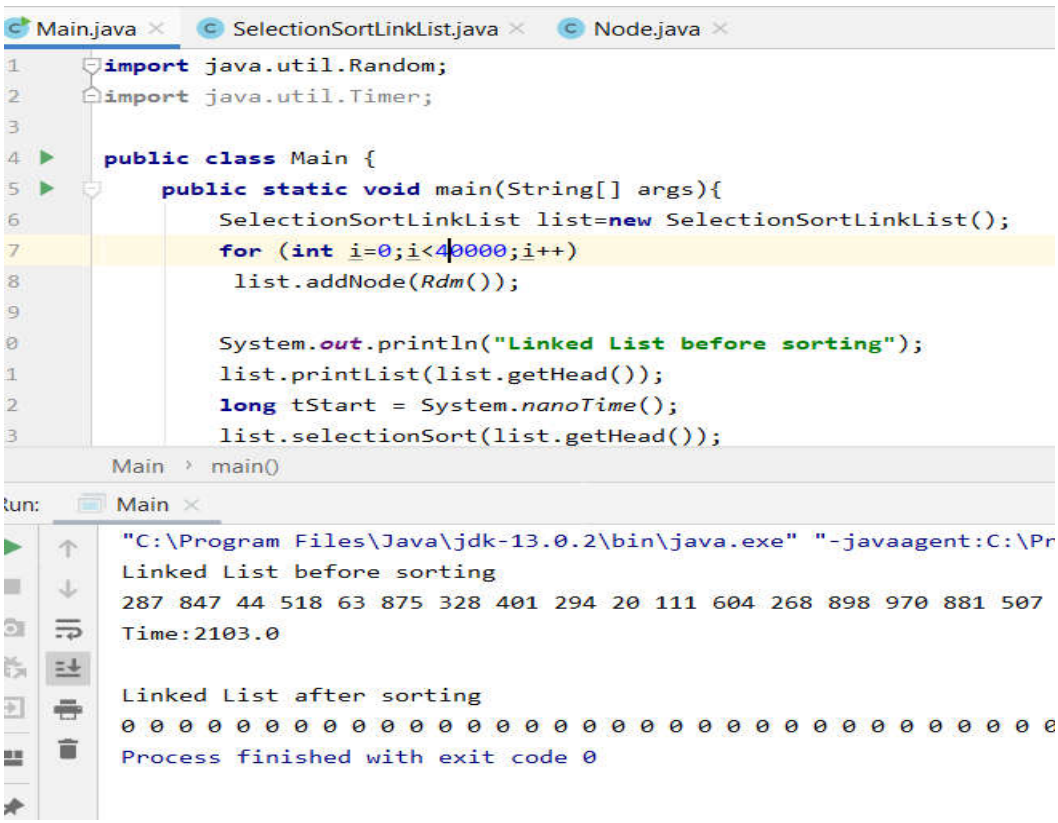
```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetB
Linked List before sorting
258 386 152 878 261 852 931 18 889 380 374 517 597 184 31 719 246 393 958 232 337
Time:0.0

Linked List after sorting
18 31 37 47 65 80 80 89 90 118 130 134 140 152 168 169 180 184 188 204 207 232 24
Process finished with exit code 0
```

B) SelectionSort-10000



```
Main.java | SelectionSortLinkList.java | Node.java

import java.util.Random;
import java.util.Timer;

public class Main {
    public static void main(String[] args){
        SelectionSortLinkList list=new SelectionSortLi
        for (int i=0;i<10000;i++)
         list.addNode(Rdm());

        System.out.println("Linked List before sorting
        list.printList(list.getHead());
        long tStart = System.nanoTime();
        list.selectionSort(list.getHead());

Main › main()

Main

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-jav
Linked List before sorting
410 438 158 847 71 717 774 82 469 231 10 451 623 695
Time:177.0

Linked List after sorting
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1
Process finished with exit code 0
```

C) SelectionSort-40000



```
Main.java | SelectionSortLinkList.java | Node.java

1  import java.util.Random;
2  import java.util.Timer;
3
4  public class Main {
5      public static void main(String[] args){
6          SelectionSortLinkList list=new SelectionSortLinkList();
7          for (int i=0;i<40000;i++)
8           list.addNode(Rdm());
9
0          System.out.println("Linked List before sorting");
1          list.printList(list.getHead());
2          long tStart = System.nanoTime();
3          list.selectionSort(list.getHead());

Main › main()

Run: Main

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Pr
Linked List before sorting
287 847 44 518 63 875 328 401 294 20 111 604 268 898 970 881 507
Time:2103.0

Linked List after sorting
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Process finished with exit code 0
```

D) SelectionSort-100000

```
    public static void main(String[] args){
        SelectionSortLinkList list=new SelectionSortLinkList();
        for (int i=0;i<100000;i++)
         list.addNode(Rdm());

        System.out.println("Linked List before sorting");
        list.printList(list.getHead());
        long tStart = System.nanoTime();
        list.selectionSort(list.getHead());
        long tEnd = System.nanoTime();
         Time(tEnd,tStart);
        System.out.println("\nLinked List after sorting");
        list.printList(list.getHead());
```

Main › main()

Main

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Fi
Linked List before sorting
409 23 815 49 304 14 999 546 886 926 571 105 178 965 989 295 449 952 610
Time:11292.0

Linked List after sorting
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Process finished with exit code 0
```

E) SelectionSort-150000

Main.java | SelectionSortLinkList.java | Node.java

```
5     public static void main(String[] args){
6         SelectionSortLinkList list=new SelectionSortLinkList();
7         for (int i=0;i<150000;i++)
8          list.addNode(Rdm());
9
10        System.out.println("Linked List before sorting");
11        list.printList(list.getHead());
12        long tStart = System.nanoTime();
13        list.selectionSort(list.getHead());
14        long tEnd = System.nanoTime();
15         Time(tEnd,tStart);
16        System.out.println("\nLinked List after sorting");
17        list.printList(list.getHead());
```

Main › main()

Run: Main

```
69 417 971 196 67 255 813 506 272 354 278 801 534 215 186 615 35 204 546 337 359 26
Time:22157.0

Linked List after sorting
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Process finished with exit code 0
```

F) SelectionSort-180000