**Client.java**

import java.util.Random;

/\*\*

\* @author joe

\*/

public class Client {

public static final int SORT\_BUBBLE\_SIMPLE = 0;

public static final int SORT\_BUBBLE\_ENHANCED = 1;

public static final int SORT\_INSERTION = 2;

public static final int SORT\_SELECTION = 3;

public static void printArray(int[] arr, int num) {

//System.out.println("Array size: " + arr.length);

for(int i = 0; i < num && i < arr.length; i++) {

System.out.println(arr[i]);

}

//System.out.println("\n");

}

public static int indexOfLargestElement(int[] arr, int size) {

int index = 0;

for(int i = 1; i < size; i++) {

if(arr[i] > arr[index])

index = i;

}

return index;

}

public static long testSortAlgorithm(int[] arr, int sort\_algorithm) {

Random r = new Random();

for(int i = 0; i < arr.length; i++) {

arr[i] = r.nextInt();

}

printArray(arr, 0);

long start\_time = System.nanoTime();

switch(sort\_algorithm) {

case SORT\_BUBBLE\_SIMPLE:

simpleBubbleSort(arr);

break;

case SORT\_BUBBLE\_ENHANCED:

enhancedBubbleSort(arr);

break;

case SORT\_INSERTION:

insertionSort(arr);

break;

case SORT\_SELECTION:

selectionSort(arr);

break;

default:

break;

}

printArray(arr, 0);

return System.nanoTime() - start\_time;

}

public static double seconds(long nanoseconds) {

double nano\_s = (double)nanoseconds;

return nano\_s / 1000000000.0;

}

public static void main(String[] args) {

for(int i = 10; i <= 100000; i \*= 10) {

int[] tmp\_arr = new int[i];

long t1 = testSortAlgorithm(tmp\_arr, SORT\_BUBBLE\_SIMPLE);

long t2 = testSortAlgorithm(tmp\_arr, SORT\_BUBBLE\_ENHANCED);

long t3 = testSortAlgorithm(tmp\_arr, SORT\_INSERTION);

long t4 = testSortAlgorithm(tmp\_arr, SORT\_SELECTION);

// help garbage collector

tmp\_arr = null;

System.out.println("Elements: " + i);

System.out.println("Simple Bubble Sort time: " + seconds(t1) + " seconds");

System.out.println("Enhanced Bubble Sort time: " + seconds(t2) + " seconds");

System.out.println("Insertion Sort time: " + seconds(t3) + " seconds");

System.out.println("Selection Sort time: " + seconds(t4) + " seconds\n");

}

}

public static void selectionSort(int[] arr) {

for(int i = 0; i < arr.length; i++) {

int max = indexOfLargestElement(arr, arr.length - i);

// swap the two elements

int temp = arr[max];

arr[max] = arr[arr.length - (i+1)];

arr[arr.length - (i+1)] = temp;

}

}

public static void insertionSort(int[] arr) {

for(int i = 0; i < arr.length; i++) {

int j = i;

int temp = arr[i];

while(j != 0 && arr[j-1] > temp) {

arr[j] = arr[j-1];

j--;

}

arr[j] = temp;

}

}

public static void enhancedBubbleSort(int[] arr) {

for(int i = 0; i < arr.length-1; i++) {

boolean inner\_swapped = false;

for(int j = 0; j < arr.length-(1+i); j++) {

if(arr[j] > arr[j+1]) {

int t = arr[j];

arr[j] = arr[j+1];

arr[j+1] = t;

inner\_swapped = true;

}

}

if(!inner\_swapped)

return;

}

}

public static void simpleBubbleSort(int[] arr) {

for(int i = 0; i < arr.length-1; i++) {

for(int j = 0; j < arr.length-1; j++) {

if(arr[j] > arr[j+1]) {

int t = arr[j];

arr[j] = arr[j+1];

arr[j+1] = t;

}

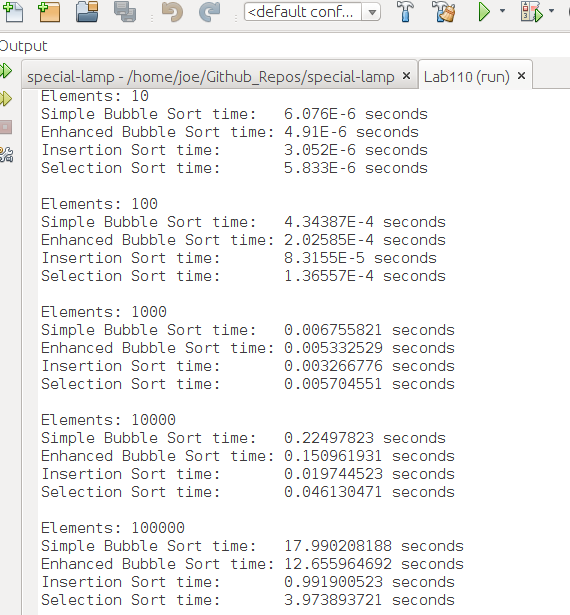
}

}

}

}

**TRUNCATED OUTPUT**:



**VERBOSE OUTPUT (11 pages):**

Array size: 10

2095442872

1737181277

1651063596

949565276

-63909749

-2098426562

1145462213

-2065254287

304049243

1072351270

Array size: 10

-2098426562

-2065254287

-63909749

304049243

949565276

1072351270

1145462213

1651063596

1737181277

2095442872

Array size: 10

1218354085

1330435448

-698294499

-387627077

627888807

-328884886

-1130982168

6602876

-1695157057

-1117041082

Array size: 10

-1695157057

-1130982168

-1117041082

-698294499

-387627077

-328884886

6602876

627888807

1218354085

1330435448

Array size: 10

-643149000

-2098979227

-187771438

-521216182

-707535911

519730655

-1517398415

-148587179

-860598947

-1927378915

Array size: 10

-2098979227

-1927378915

-1517398415

-860598947

-707535911

-643149000

-521216182

-187771438

-148587179

519730655

Array size: 10

-1210473722

-744336357

-188551933

449903863

458876732

-2128657916

-1036835449

-1422932015

-1170091153

652612759

Array size: 10

-2128657916

-1422932015

-1210473722

-1170091153

-1036835449

-744336357

-188551933

449903863

458876732

652612759

Elements: 10

Simple Bubble Sort time: 3.21899E-4 seconds

Enhanced Bubble Sort time: 5.30687E-4 seconds

Insertion Sort time: 5.3974E-4 seconds

Selection Sort time: 5.73327E-4 seconds

Array size: 100

-1488048512

1571682795

720240149

-656509206

1844454118

951933626

-1708043917

-412412090

540177829

1553516095

Array size: 100

-2117759146

-2101427026

-2073894504

-2050783414

-2044982463

-2013075698

-1973927427

-1957607630

-1951976137

-1878587641

Array size: 100

-670762909

2051796853

1341345026

-1762082549

-375428939

1296363699

2011188002

1596921341

-524794729

674435152

Array size: 100

-2101771789

-2100760846

-2085272040

-2068007369

-2038417863

-1970987909

-1923753380

-1905059670

-1876873149

-1842786626

Array size: 100

-442166564

321306174

1371714948

-167771699

-240447944

-2143161868

-1057826743

-2010395386

-1438139159

1011307874

Array size: 100

-2143161868

-2126254988

-2083253256

-2032486626

-2010395386

-1953734042

-1946961382

-1893765153

-1891602152

-1853491380

Array size: 100

970570898

-1943079186

898056495

2035326867

-1868614532

-1944617459

-1542545870

35860442

2108173715

34575890

Array size: 100

-2076646202

-1975124746

-1944617459

-1943079186

-1932677325

-1897834486

-1891323883

-1868614532

-1798329230

-1778832305

Elements: 100

Simple Bubble Sort time: 6.68731E-4 seconds

Enhanced Bubble Sort time: 8.32909E-4 seconds

Insertion Sort time: 3.15828E-4 seconds

Selection Sort time: 6.35142E-4 seconds

Array size: 1000

1795219707

2082697519

-672888079

414363413

-1403556025

-543692891

1233974615

1835263137

1665708287

-191237833

Array size: 1000

-2143419995

-2141135172

-2130617160

-2127091815

-2125199766

-2121947845

-2118686852

-2117913713

-2114372042

-2113671150

Array size: 1000

-1594544459

1562608639

-1647492639

-2142046425

1327200492

-758149983

894031252

-97337181

1234240120

-866760935

Array size: 1000

-2147277379

-2146842920

-2144825033

-2144133419

-2142046425

-2141465282

-2132059112

-2120961294

-2120760137

-2119241210

Array size: 1000

-2086369705

-1147308638

-1405882494

-587807318

1098261910

1237697338

503937703

-2142939121

227379735

1381257371

Array size: 1000

-2145246420

-2143566648

-2142939121

-2136761006

-2134337080

-2128676430

-2116055712

-2113008527

-2109411606

-2106810832

Array size: 1000

598325331

-1917096729

2069830672

-2091900902

299205859

-1432930339

-1330985871

1637238176

-461561991

-1340832205

Array size: 1000

-2143938731

-2139213570

-2138872052

-2136903108

-2134900586

-2132727686

-2132091648

-2130652768

-2122986228

-2120265287

Elements: 1000

Simple Bubble Sort time: 0.007082872 seconds

Enhanced Bubble Sort time: 0.006601102 seconds

Insertion Sort time: 0.003747266 seconds

Selection Sort time: 0.00255205 seconds

Array size: 10000

842248391

411570297

-430974053

1177102533

-1695610097

-1757050851

1107827440

808676671

-69573128

1034905659

Array size: 10000

-2147208025

-2147154732

-2147114767

-2146406806

-2146110339

-2145845220

-2145388223

-2145182517

-2144944625

-2144896817

Array size: 10000

-1413899072

140272138

-278544279

1192498156

593504221

-2118610825

-204527750

2111654641

-572704991

1625276362

Array size: 10000

-2147015507

-2145766979

-2145627117

-2145473236

-2145405162

-2145160639

-2143791726

-2143215275

-2142388296

-2140816919

Array size: 10000

1707785259

-1128392080

757270691

-1921990955

-882494740

-567100532

570457753

-1419886326

-1129801276

-1864811112

Array size: 10000

-2147268303

-2147176891

-2147120728

-2146979969

-2146787735

-2146698837

-2146237637

-2145953588

-2145805744

-2145702592

Array size: 10000

-1403992347

-1976339150

388210256

963029887

742933357

1827956588

-770758503

-973672025

-1738091325

2008945817

Array size: 10000

-2147314722

-2147026172

-2146955911

-2146530501

-2146451682

-2146266403

-2145933595

-2144914732

-2143854974

-2143473891

Elements: 10000

Simple Bubble Sort time: 0.221710196 seconds

Enhanced Bubble Sort time: 0.137444385 seconds

Insertion Sort time: 0.010533022 seconds

Selection Sort time: 0.042033894 seconds

Array size: 100000

1223378135

2142460104

-137964249

-1854207265

556694230

1525378087

-1068016360

-576292404

759152509

-1054201094

Array size: 100000

-2147475278

-2147435703

-2147406245

-2147250647

-2147212299

-2147159083

-2147138871

-2146971421

-2146938304

-2146909447

Array size: 100000

309121133

-1785475252

204899111

-1344116621

1724130114

1062830346

1900899377

-1632243546

2051530289

541545411

Array size: 100000

-2147434912

-2147272834

-2147272142

-2147259120

-2147246943

-2147229564

-2147186741

-2147128778

-2147032328

-2146991148

Array size: 100000

-2058602626

-2036775685

-181887757

-357849909

-2009423550

-680946382

-1195922409

124540374

1725095110

1215803334

Array size: 100000

-2147422512

-2147395224

-2147248927

-2147226385

-2147181148

-2147140896

-2147136932

-2147115847

-2147014592

-2147010736

Array size: 100000

-261675553

913379505

-1899643177

1979564594

-1044443998

1147859525

1822081098

-1189682220

-825880957

-2097437710

Array size: 100000

-2147474926

-2147474452

-2147469752

-2147462107

-2147321341

-2147306863

-2147238473

-2147200311

-2147175590

-2147166956

Elements: 100000

Simple Bubble Sort time: 17.8156726 seconds

Enhanced Bubble Sort time: 13.143975553 seconds

Insertion Sort time: 0.953314576 seconds

Selection Sort time: 3.84631268 seconds