Министерство цифрового развития, связи и массовых коммуникаций Государственное образовательного учреждение высшего образования

Ордена Трудового Красного Знамени

«Московский технический университет связи и информатики»

Лабораторная работа № 1

«Методы сортировки»

Выполнил студент

группы БВТ1902

Гинатуллин Айгиз

Москва

2021

# Задание

Реализовать методы сортировок строк числовой матрицы. Оценить время работы каждого алгоритма сортировки.

Сортировки:

* Выбором
* Вставкой
* Обменом
* Шелла
* Быстрая
* Пирамидальная

# Выполнение

Листинг программы:

Lab1.js

let m = 50,n =50,minLimit = -250,maxLimit = 1010  
let array = new Array()  
  
for(let i = 0;i < ***n***;i++){  
 ***array***[i] = new ***Array***()  
 for(let j = 0;j < ***m***;j++){  
 ***array***[i][j]=***Math***.random()\*(maxLimit-minLimit)+minLimit  
 }  
}  
  
const Action = (array,action)=>{  
 const SelectionSort = (A)=>{  
  
 var n = A.length;  
 for (var i = 0; i < n-1; i++) {  
 var min = i;  
 for (var j = i+1; j < n; j++) {  
 if (A[j] < A[min]) min = j;  
 }  
 var t = A[min];  
 A[min] = A[ i ];  
 A[ i ] = t;  
 }  
 return A;  
 }  
 const InsertionSort = (A) =>{  
 var n = A.length;  
 for (var i = 0; i < n; i++) {  
 var v = A[ i ], j = i-1;  
 while (j >= 0 && A[j] > v) {  
 A[j+1] = A[j];  
 j--;  
 }  
 A[j+1] = v;  
 }  
 return A;  
 }  
 const BubbleSort = (A) =>{  
 var n = A.length;  
 for (var i = 0; i < n-1; i++) {  
 for (var j = 0; j < n-1-i; j++) {  
 if (A[j+1] < A[j]) {  
 var t = A[j+1]; A[j+1] = A[j]; A[j] = t;  
 }  
 }  
 }  
 return A;  
 }  
 const ShellSort = (A) =>{  
 var n = A.length, i = Math.floor(n/2);  
 while (i > 0)  
 { for (var j = 0; j < n; j++)  
 { var k = j, t = A[j];  
 while (k >= i && A[k-i] > t)  
 { A[k] = A[k-i]; k -= i; }  
 A[k] = t;  
 }  
 i = (i===2) ? 1 : Math.floor(i\*5/11);  
 }  
 return A;  
 }  
 const QuickSort = (A)=> {  
 if (A.length === 0) return [];  
 var a = [], b = [], p = A[0];  
 for (var i = 1; i < A.length; i++) {  
 if (A[ i ] < p) a[a.length] = A[ i ];  
 else b[b.length] = A[ i ];  
 }  
 return QuickSort(a).concat( p,QuickSort(b) );  
 }  
 function swap(a, i, j) {  
 var tmp = a[i];  
 a[i] = a[j];  
 a[j] = tmp;  
 }  
  
 function max\_heapify(a, i, length) {  
 while (true) {  
 var left = i\*2 + 1;  
 var right = i\*2 + 2;  
 var largest = i;  
  
 if (left < length && a[left] > a[largest]) {  
 largest = left;  
 }  
  
 if (right < length && a[right] > a[largest]) {  
 largest = right;  
 }  
  
 if (i === largest) {  
 break;  
 }  
  
 swap(a, i, largest);  
 i = largest;  
 }  
 }  
  
 function heapify(a, length) {  
 for (var i = Math.floor(length/2); i >= 0; i--) {  
 max\_heapify(a, i, length);  
 }  
 }  
  
 function heapsort(a) {  
 heapify(a, a.length);  
  
 for (var i = a.length - 1; i > 0; i--) {  
 swap(a, i, 0);  
  
 max\_heapify(a, 0, i-1);  
 }  
 }  
  
  
 switch (action){  
 case "selection":{  
 array = SelectionSort(array)  
 break  
 }  
 case "insertion":{  
 array = InsertionSort(array)  
 break  
 }  
 case "bubble":{  
 array = BubbleSort(array)  
 break  
 }  
 case "Shell":{  
 array = ShellSort(array)  
 break  
 }  
 case "quick":{  
 array = QuickSort(array)  
 break  
 }  
 case "heap":{  
 heapsort(array)  
 break  
 }  
 default:{  
 array = QuickSort(array)  
 }  
 }  
 console.log(array)  
  
}  
  
  
  
Action(array[0],"selection")  
Action(array[1],"insertion")  
Action(array[2],"bubble")  
Action(array[3],"Shell")  
Action(array[4],"quick")  
Action(array[4],"heap")

экран выполнения программы

[  
 -188.5581650948679, -171.70150822451905, -144.16377330318662,  
 -143.5341487040738, -58.75090447392188, -46.84728193941038,  
 -35.36210407419577, -32.15595180471155, 2.7547007510054016,  
 20.631492437944814, 43.212871076520344, 68.72196340830055,  
 86.25412987401091, 90.74548139647675, 102.50889627520161,  
 115.44020383455484, 141.06371401065041, 146.5548726474484,  
 150.81214720402858, 176.7772625883744, 189.35691896453494,  
 195.7759228055752, 226.96813798347858, 250.82796532784778,  
 258.1055786495746, 289.43040757110134, 326.85918260277833,  
 436.5351009126754, 462.8204432545192, 506.6792138793347,  
 514.0794105990502, 546.3929332997703, 557.4249855368386,  
 583.5795768552689, 704.7411938539684, 715.3319842189069,  
 739.4581087937887, 799.2120272100756, 805.374361651061,  
 826.2829074051449, 826.8628672244265, 827.782049074113,  
 849.9480614006682, 873.4298593593692, 892.8516093229869,  
 901.388246459507, 927.9521528969806, 948.2602389463375,  
 954.4053159829439, 954.7854757323375  
]  
 [  
 -244.71373416815365, -229.86314716546966, -146.05410465183635,  
 -123.56955524983077, -75.87425473481875, -55.00587774852323,  
 -49.921830019499396, -39.963726477436154, -9.85901612683952,  
 18.44536873996543, 33.6014654052716, 34.22159519604543,  
 42.14641282874072, 44.26099115290191, 77.95795345885466,  
 143.04675606057305, 177.09606204539875, 184.52819304621187,  
 198.98568804178643, 243.96699441752298, 247.5171883982219,  
 352.3852592991748, 354.86526345740845, 362.3180544394785,  
 388.97927902824165, 399.95051496772214, 400.74597704133714,  
 444.8503380842873, 471.27901545556927, 503.35935191312797,  
 522.4572949311427, 522.7948555210564, 565.264237547163,  
 588.9459868294024, 621.5750858850884, 657.2182256737464,  
 657.9908886962189, 669.2921126637247, 785.3238305267403,  
 797.1065415720752, 800.2158492828901, 846.3194304782464,  
 868.7179690970959, 909.5369671016101, 922.862476542557,  
 937.1801837374642, 943.4146279898596, 978.3515937081729,  
 978.7641476300687, 991.7221456708633  
 ]  
 [  
 -246.8761757105372, -218.97407683607645, -212.86096433799196,  
 -207.42498789629443, -142.1567605694167, -123.83854615709794,  
 -107.28428359649931, -107.1454206303905, -67.56531252004271,  
 -58.04071262481315, -42.27309748471296, 43.257338023948535,  
 85.36157230861733, 85.63488732816603, 113.8102477289583,  
 150.5558490368869, 165.0661691774116, 208.10895838750042,  
 211.9299813931607, 232.6860146836105, 239.10776275447228,  
 381.51127507146737, 386.1570611627792, 414.95945546805683,  
 450.41238997387063, 461.75936341582917, 468.5334832799358,  
 474.3868669993726, 522.8318074798092, 526.6848737591495,  
 530.2835470968333, 537.2482840690701, 562.3717348191294,  
 626.6319243208268, 641.8654413804024, 758.0660199097298,  
 765.7900723494095, 811.9318134086325, 813.8520213921922,  
 814.965931158994, 848.083522733087, 885.4611131281933,  
 887.1928744861489, 898.3278482471026, 929.9200740628662,  
 946.726029248565, 948.6191733424639, 949.4463747847476,  
 969.4785421852096, 985.9588125670148  
 ]  
 [  
 -239.9023283009954, -160.72555439647113, -157.6426153717968,  
 -55.85197785304888, -47.66505914625202, -26.469515434772006,  
 1.6402965370841969, 68.99357000511884, 70.5233625079465,  
 103.35838859291346, 178.6710704792792, 183.93578224237007,  
 197.27707239589404, 215.84062957478494, 220.1424294791866,  
 296.187263073116, 328.4928331873542, 341.38213244754843,  
 345.86859315965614, 353.3753804324998, 361.22342696446765,  
 366.81227710245696, 377.2708243479457, 387.21474873419277,  
 407.0273967465856, 417.1827507682973, 429.0736678921054,  
 437.1816053625116, 472.3923167146264, 482.35781525655443,  
 487.8479775647122, 507.8842941823033, 524.5923936841995,  
 553.6796872383425, 557.4907240218556, 567.0405400305795,  
 570.6025435576321, 628.509368337623, 657.0948184854745,  
 670.3053936311588, 771.2214167647078, 862.8683280403905,  
 887.8311280833345, 910.3997084636642, 916.9851772492348,  
 921.9528608993676, 934.8579232808313, 938.1129324719093,  
 965.3541638931572, 1007.295735718164  
 ]  
 [  
 -248.81295197815467, -245.1085049959087, -223.28941235857113,  
 -192.88623551443285, -151.94931822649312, -148.84744746086193,  
 -118.54734417070097, -97.04564136110014, -73.58172338093942,  
 -65.37387516308257, -43.167664705645166, 20.51572885135272,  
 95.12951799175278, 112.88666661954261, 160.06513250721798,  
 160.3401655737369, 164.93585452594203, 203.03425903210496,  
 221.82370396033036, 225.87821154606735, 242.58184560757957,  
 259.52218111367887, 322.8427310429536, 337.3884570868861,  
 339.80438333221457, 344.70679209301466, 363.7355725571357,  
 414.3251928295949, 415.624700129179, 440.99798291128025,  
 460.9772472410941, 474.5255493952461, 531.9001068850758,  
 556.4149045139807, 572.2266159803544, 646.1353158239151,  
 658.8791988775507, 662.2184573766959, 680.4228703330276,  
 697.4453125404134, 725.1184865151887, 748.2220548579508,  
 755.9767243371002, 771.2371057788223, 782.8354850461637,  
 798.3767086256917, 933.5607356464586, 981.1604512834401,  
 986.6653052104868, 989.5652273446105  
 ]  
 [  
 -248.81295197815467, -245.1085049959087, -223.28941235857113,  
 -192.88623551443285, -151.94931822649312, -148.84744746086193,  
 -118.54734417070097, -97.04564136110014, -73.58172338093942,  
 -65.37387516308257, -43.167664705645166, 20.51572885135272,  
 95.12951799175278, 112.88666661954261, 160.06513250721798,  
 160.3401655737369, 164.93585452594203, 203.03425903210496,  
 221.82370396033036, 225.87821154606735, 242.58184560757957,  
 259.52218111367887, 322.8427310429536, 337.3884570868861,  
 339.80438333221457, 344.70679209301466, 363.7355725571357,  
 414.3251928295949, 415.624700129179, 440.99798291128025,  
 460.9772472410941, 474.5255493952461, 531.9001068850758,  
 556.4149045139807, 572.2266159803544, 646.1353158239151,  
 658.8791988775507, 662.2184573766959, 680.4228703330276,  
 697.4453125404134, 725.1184865151887, 748.2220548579508,  
 755.9767243371002, 771.2371057788223, 782.8354850461637,  
 798.3767086256917, 933.5607356464586, 981.1604512834401,  
 986.6653052104868, 989.5652273446105  
 ]

Рис. 1 Исходная матрица

# Вывод

В ходе выполнения лабораторной работы, я реализовал сортировки выбором, вставками, обменом, Шелла, быстрой и пирамидальной.