**Отчет по заданию №1 по практикуму на ЭВМ: «Методы сортировки. Язык Паскаль».**

**Автор – Левченко В. В., группа 141**

**Задача программы**:   
Реализовать два метода сортировки массива целых чисел (метод пузырьком и метод Шелла, сортировка по невозрастанию) и провести их экспериментальное сравнение. Конкретные методы и вид сортировки определяются вариантом задания. Для каждого из реализуемых методов необходимо предусмотреть возможность работы с массивами длины от до ().

При реализации каждого метода вычислить число сравнений элементов и число перемещений элементов (либо обменов пар элементов, либо пересылок элементов на новые места - в зависимости от метода), выполненных в процессе сортировки.

**Код программы (язык Pascal):**

const

m = 100;

var

cmp, swp: integer;

type

ar = array[1..m] of integer;

procedure generate(p: integer; N: integer; var a: ar); //генерирует случайный массив, от параметра p зависит то, в каком порядке массив, который записывается в a

var i: integer;

begin

randomize;

a[1] := random(32000);

if p = 1 then

for i := 2 to N do

a[i] := a[i-1] - random(32000);

if p = 2 then

for i := 2 to N do

a[i] := a[i-1] + random(32000);

if p > 2 then

for i := 2 to N do

a[i] := random(32000);

end;

procedure sort\_bubble(N: integer; var x: ar); //сортировка пузырьком, N – кол-во элементов, записываемых в x

var i, j, k: integer;

begin

for i := 1 to N-1 do

for j := 1 to N-i do begin

if x[j] < x[j+1] then begin //если элемент меньше следующего, меняем местами

k := x[j];

x[j] := x[j+1];

x[j+1] := k;

swp := swp + 1;

end;

cmp := cmp + 1;

end;

end;

procedure sort\_Shell(N: integer; var x: ar); //сортировка Шелла, N – кол-во элементов, записываемых в x

var

s, i, j, k: integer;

begin

s := N div 2;

while s > 0 do begin

for i := 1 to N do begin

j := i+s; //смотрим на элементы, отличающиеся на s позиций

while j <= N do begin

if x[i] < x[j] then begin

k := x[j]; //если i-й элемент меньше j-го, меняем местами

x[j] := x[i];

x[i] := k;

swp := swp + 1;

end;

cmp := cmp + 1;

j := j+s; //переходим на следующий элемент через s позиций

end;

end;

s := s div 2; //уменьшаем s

end;

end;

var

N, i: integer;

a: ar;

begin

cmp := 0;

swp := 0;

readln(N);

generate(2, N, a);

for i := 1 to N do write(a[i], ' ');

writeln;

//sort\_bubble(N, a);

sort\_Shell(N, a);

for i := 1 to N do write(a[i], ' ');

writeln;

write(cmp, ' ', swp);

end.

**Метод сортировки пузырьком**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **n** | **Параметр** | **Номер сгенерированного массива** | | | | **Среднее значение** |
| **1 (элементы уже упорядочены)** | **2 (элементы упорядочены в обратном порядке)** | **3 (случайный)** | **4 (случайный)** |
| **10** | Сравнения | 45 | 45 | 45 | 45 | **45** |
| Перемещения | 0 | 45 | 16 | 18 | **19.75** |
| **20** | Сравнения | 190 | 190 | 190 | 190 | **190** |
| Перемещения | 0 | 190 | 83 | 101 | **93.5** |
| **50** | Сравнения | 1225 | 1225 | 1225 | 1225 | **1225** |
| Перемещения | 0 | 1225 | 675 | 710 | **652.5** |
| **100** | Сравнения | 4950 | 4950 | 4950 | 4950 | **4950** |
| Перемещения | 0 | 4950 | 2789 | 2014 | **2438.25** |

Наихудший случай - число сравнений и обменов равно .

Наилучший случай – число сравнений равно , обменов 0.

**Метод сортировки Шелла**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **n** | **Параметр** | **Номер сгенерированного массива** | | | | **Среднее значение** |
| **1** | **2** | **3** | **4** |
| **10** | Сравнения | 70 | 70 | 70 | 70 | **70** |
| Перемещения | 0 | 13 | 14 | 8 | **8.75** |
| **20** | Сравнения | 320 | 320 | 320 | 320 | **320** |
| Перемещения | 0 | 36 | 34 | 43 | **28.25** |
| **50** | Сравнения | 1906 | 1906 | 1906 | 1906 | **1906** |
| Перемещения | 0 | 105 | 151 | 155 | **102.75** |
| **100** | Сравнения | 7919 | 7919 | 7919 | 7919 | **7919** |
| Перемещения | 0 | 260 | 395 | 389 | **261** |

Даже в наихудшем случае обменов будет меньше, чем , благодаря поэтапному сокращению, происходящему внутри алгоритма.

Наилучший случай – сравнений столько же, сколько при сортировкой пузырьком, обменов 0.

**Тестирование:**

**n = 10:**

1. 10 … 1 -> 10 … 1
2. 1 .. 10 -> 10 … 1
3. -349 434 274 200 186 168 -462 -76 312 -251 -> 434 312 274 200 186 168 -76 -251 -349 -462
4. 175 312 -187 435 -190 -130 151 -306 -397 491 -> 491 435 312 175 151 -130 -187 -190 -306 -397

**n = 20:**

1. 20 … 1 -> 20 … 1
2. 1 … 20 -> 20 … 1
3. -349 434 274 200 186 168 -462 -76 312 -251 175 312 -187 435 -190 -130 151 -306 -397 491 -> 491 435 434 312 312 274 200 186 175 168 151 -76 -130 -187 -190 -251 -306 -349 -397 -462
4. -224 -482 47 221 -445 30 -450 -50 186 297 -462 -81 427 -220 -152 292 -4 24 -198 -383 224 -> 491 435 434 312 312 274 200 186 175 168 151 -76 -130 -187 -190 -251 -306 -349 -397 -462

**n = 50:**

1. 50 … 1 -> 50 … 1
2. 1 … 50 -> 50 … 1
3. -182 -166 174 -356 -24 -464 141 449 -283 -91 181 122 32 88 20 -411 293 -181 -436 47 184 128 -222 369 196 -277 -376 400 323 165 408 -309 17 -131 -208 255 80 233 241 114 -276 -456 341 306 376 256 -271 -70 -294 -16 -> 449 408 400 376 369 341 323 306 293 256 255 241 233 196 184 181 174 165 141 128 122 114 88 80 47 32 20 17 -16 -24 -70 -91 -131 -166 -181 -182 -208 -222 -271 -276 -277 -283 -294 -309 -356 -376 -411 -436 -456 -464
4. -303 61 27 -383 -475 -11 376 -20 -347 -174 -34 -360 -48 -163 335 -250 -500 -55 -453 -253 492 396 -425 -284 421 0 156 387 -275 228 198 -238 -11 263 -340 -444 260 364 -233 -286 496 446 276 -222 303 -75 312 -266 442 -290 -> 496 492 446 442 421 396 387 376 364 335 312 303 276 263 260 228 198 156 61 27 0 -11 -11 -20 -34 -48 -55 -75 -163 -174 -222 -233 -238 -250 -253 -266 -275 -284 -286 -290 -303 -340 -347 -360 -383 -425 -444 -453 -475 -500

**n = 100:**

1. 100 … 1 -> 100 … 1
2. 1 … 100 -> 1 … 100
3. 42 -93 -328 -245 86 196 -254 169 -340 -347 398 -320 416 -121 -411 457 366 -415 176 -96 -35 313 443 -419 -294 306 63 250 -197 -85 -137 464 106 -287 492 -273 -224 -292 250 -388 -230 -1 -113 222 415 22 -407 168 339 83 -486 13 338 290 -199 228 -36 -409 -452 279 -368 -4 -135 245 289 433 74 203 -114 362 181 157 -166 63 -367 466 -75 -279 -334 184 213 395 410 219 -100 350 -375 316 412 -237 262 -315 -340 471 480 273 -367 142 433 481 -> 492 481 480 471 466 464 457 443 433 433 416 415 412 410 398 395 366 362 350 339 338 316 313 306 290 289 279 273 262 250 250 245 228 222 219 213 203 196 184 181 176 169 168 157 142 106 86 83 74 63 63 42 22 13 -1 -4 -35 -36 -75 -85 -93 -96 -100 -113 -114 -121 -135 -137 -166 -197 -199 -224 -230 -237 -245 -254 -273 -279 -287 -292 -294 -315 -320 -328 -334 -340 -340 -347 -367 -367 -368 -375 -388 -407 -409 -411 -415 -419 -452 -486
4. -25 -23 345 430 -409 -264 -345 14 225 251 -18 91 493 160 65 -377 -491 -286 -496 175 268 -106 461 189 398 281 -53 491 -20 160 -119 -54 -135 -49 -417 -166 -52 221 78 345 291 414 -22 1 -265 11 -97 -108 336 -164 -335 -298 -141 -232 -343 -419 15 -344 -469 -190 289 -336 -467 -436 -131 -465 -275 41 77 -269 175 -212 136 -342 384 354 415 -40 -47 -114 -123 -446 -346 -356 -5 -322 -413 -404 -206 -499 318 -208 189 -232 -444 126 380 -86 -369 -301 -> 493 491 461 430 415 414 398 384 380 354 345 345 336 318 291 289 281 268 251 225 221 189 189 175 175 160 160 136 126 91 78 77 65 41 15 14 11 1 -5 -18 -20 -22 -23 -25 -40 -47 -49 -52 -53 -54 -86 -97 -106 -108 -114 -119 -123 -131 -135 -141 -164 -166 -190 -206 -208 -212 -232 -232 -264 -265 -269 -275 -286 -298 -301 -322 -335 -336 -342 -343 -344 -345 -346 -356 -369 -377 -404 -409 -413 -417 -419 -436 -444 -446 -465 -467 -469 -491 -496 -499

**Вывод** – по количеству сравнений метод пузырька более эффективный, однако метод Шелла более эффективен по количеству перемещений, а значит и по времени, занимаемому выполнением программы.