Міністерство освіти і науки України

Національний університет “Одеська політехніка”

Інститут комп’ютерних систем

Кафедра інформаційних систем

Лабораторна робота №1

З дисципліни: «Теорія Алгоритмів»

Тема: «Квадратичні алгоритми сортування»

Варіант №20

Виконала:

Студент групи АІ-223

Перегудов Егор Володимирович

Перевірили:

Арсірій О. О.

Одеса 2023

Завдання 1.Відсортувати за допомогою сортування вибором

for i <-- 0 to n-2 do

min <-- i

for j <-- i+1 to n-1 do

if A[j] < A[min]

min <-- j

Обмін А[i] та A [min]

1.Вихидні дані А=[70, 891, 42, 683, 704, 495, 106, 627, 518]

1.1 i=0; min=0; j=0+1=1; A[1]<A[0] (89<7)min=0

1.2 j=2; A[2]<A[0] (4<7)min=2

1.3 j=3; A[3]<A[2] (68<4)min=2

1.4 j=4; A[4]<A[1] (70<4)min=2

1.5 j=5; A[5]<A[1] (49<4)min=2

1.6 j=6; A[6]<A[1] (10<4)min=2

1.7 j=7; A[7]<A[1] (62<4)min=2

1.8 j=8; A[8]<A[1] (51<4)min=2 (end for j)

b=A[0] A[0]=A[2] A[2]=b A[2]=4 A[0]=7

2 А=[40, 891, 72, 683, 704, 495, 106, 627, 518]

2.1 i=1; min=1; j=1+1=2; A[2]<A[1] (7<89)min=2

2.2 j=3; A[3]<A[2] (68<7)min=2

2.3 j=4; A[4]<A[2] (70<7)min=2

2.4 j=5; A[5]<A[2] (49<7)min=2

2.5 j=6; A[6]<A[2] (10<7)min=2

2.6 j=7; A[7]<A[2] (62<7)min=2

2.7 j=7; A[8]<A[2] (51<7)min=2

b=A[1] A[1]=A[2] A[2]=b A[1]=7 A[2]=89

3. А=[40, 71, 892, 683, 704, 495, 106, 627, 518]

3.1 i=2; min=2; j=2+1=3; A[3]<A[2] (68<89)min=3

3.2 j=4; A[4]<A[3] (70<68)min=3

3.3 j=5; A[5]<A[2] (49<68)min=5

3.4 j=6; A[6]<A[5] (10<49)min=6

3.5 j=7; A[7]<A[6] (62<10)min=6

3.6 j=8; A[8]<A[6] (51<10)min=6

b=A[2] A[2]=A[6] A[6]=b A[2]=10 A[6]=89

4. А=[40, 71, 102, 683, 704, 495, 896, 627, 518]

4.1 i=3; min=3; j=3+1=4; A[4]<A[3] (70<68)min=3

4.2 j=5; A[5]<A[3] (49<68)min=5

4.3 j=6; A[6]<A[5] (89<49)min=5

4.4 j=7; A[7]<A[5] (62<49)min=5

4.5 j=8; A[8]<A[5] (51<68)min=5

b=A[3] A[3]=A[5] A[3]=b A[3]=49 A[5]=68

5. А=[40, 71, 102, 493, 704, 685, 896, 627, 518]

5.1 i=4; min=4; j=4+1=5; A[5]<A[4] (68<70)min=5

5.2 j=6; A[6]<A[5] (89<68)min=5

5.2 j=7; A[7]<A[5] (62<68)min=7

5.2 j=8; A[8]<A[7] (51<62)min=8

b=A[4] A[4]=A[8] A[4]=b A[4]=51 A[8]=70

6. А=[40, 71, 102, 493, 514, 685, 896, 627, 708]

6.1 i=5; min=5; j=5+1=6; A[6]<A[5] (89<68)min=5

6.2 j=7; A[7]<A[5] (62<68)min=7

6.3 j=8; A[8]<A[7] (70<62)min=7

b=A[5] A[5]=A[7] A[5]=b A[5]=62 A[7]=68

7. А=[40, 71, 102, 493, 514, 625, 896, 687, 708]

7.1 i=6; min=6; j=6+1=7; A[7]<A[6] (68<89)min=7

7.2 j=8; A[8]<A[7] (70<68)min=7

b=A[6] A[6]=A[7] A[6]=b A[6]=68 A[7]=89

8. А=[40, 71, 102, 493, 514, 625, 686, 897, 708]

8.1 i=7; min=7; j=7+1=8; A[7]<A[8] (89<70)min=8 (end for j)

b=A[7] A[7]=A[8] A[7]=b A[7]=70 A[8]=89

А=[40, 71, 102, 493, 514, 625, 686, 707, 898]

(end for i)

Завдання 2: відсортувати за допомогою сортування вставками

Вихидні дані А=[71, 892, 43, 684, 705, 496, 107, 628, 519] n=9]

A.length=9

1. j = 2; key = A[2] = 89; i = 2 – 1 = 1;

i>0 (True) A[1] > key (7>89 – False) →

A[2]=key=89

А=[71, 892, 43, 684, 705, 496, 107, 628, 519]

2. j = 3; key = A[3] = 4; i = 3 – 1 = 2;

i>0 (True) A[2] > key (89>4 – True) → A[3] = A[2] =89

А=[71, 892, 893, 684, 705, 496, 107, 628, 519]

i=2-1=0

i>0 (True) A[1] > key (7>4 – True) → A[2] = A[1] =7

А=[71, 72, 893, 684, 705, 496, 107, 628, 519]

i=1-1=0

i>0 (False)

A[1] = key = 4

А=[41, 72, 893, 684, 705, 496, 107, 628, 519]

3. j=4; key = A[4] = 68 i = 4 – 1 = 3;

i>0 (True) A[3] > key (89>68 – True) → A[4] = A[3] =89

i=3-1=2

i>0 (True) A[2] > key (7>68 – False) →

A[3] = key = 68

А=[41, 72, 683, 894, 705, 496, 107, 628, 519]

4. j=5; key = A[5] = 70 i = 5 – 1 = 4;

i>0 (True) A[4] > key (89>70 – True) → A[5] = A[4] =89

i=4-1=3

i>0 (True) A[3] > key (68>70 – False) →

A[4] = key = 70

А=[41, 72, 683, 704, 895, 496, 107, 628, 519]

5. j=6; key = A[6] = 49 i = 6 – 1 = 5;

i>0 (True) A[5] > key (89>49 – True) → A[6] = A[5] = 89

i=5-1=4

i>0 (True) A[4] > key (70>49 – True) → A[5] = A[4] = 70

i=4-1=3

i>0 (True) A[3] > key (68>49 – True) → A[4] = A[3] = 68

i =3-1=2

i>0 (True) A[2] > key (7>49 – False) →

A[3]=key=49

А=[41, 72, 493, 684, 705, 896, 107, 628, 519]

6. j=7; key = A[7] = 10 i = 7 – 1 = 6;

i>0 (True) A[6] > key (89>10 – True) → A[7] = A[6] = 89

i=6-1=5

i>0 (True) A[5] > key (70>10 – True) → A[6] = A[5] = 70

i=5-1=4

i>0 (True) A[4] > key (68>10 – True) → A[5] = A[4] = 68

i=4-1=3

i>0 (True) A[3] > key (49>10 – True) → A[4] = A[3] = 49

i=3-1=2

i>0 (True) A[2] > key (7>10 – False) →

A[3]=key=10

А=[41, 72, 103, 494, 685, 706, 897, 628, 519]

7. j=8; key = A[8] = 62; i = 8 – 1 = 7;

i>0 (True) A[7] > key (89>62 – True) → A[8] = A[7] = 89

i=7-1=6

i>0 (True) A[6] > key (70>62 – True) → A[7] = A[6] = 70

i=6-1=5

i>0 (True) A[5] > key (68>62 – True) → A[6] = A[5] = 68

i=5-1=4

i>0 (True) A[4] > key (49>62 – True) →

A[5]=key=62

А=[41, 72, 103, 494, 625, 686, 707, 898, 519]

8. j=9; key = A[9] = 51; i = 9 – 1 = 8;

i>0 (True) A[8] > key (89>51 – True) → A[9] = A[8] = 89

i=8-1=7

i>0 (True) A[7] > key (70>51 – True) → A[8] = A[7] = 70

i=7-1=6

i>0 (True) A[6] > key (68>51 – True) → A[7] = A[6] = 68

i=6-1=5

i>0 (True) A[5] > key (62>51 – True) → A[6] = A[5] = 62

i=5-1=4

i>0 (True) A[4] > key (49>51 – False) →

A[5]=key=51

А=[41, 72, 103, 494, 515, 626, 687, 708, 899]

End for j

Завдання 3: порівняти між собою алгоритми сортування за кількістю операцій порівняння та присвоювання.

Складність використання обох алгоритмів та кількість кроків під час процесу сортування залежить від кількості елементів у списку. У моєму випадку, перший алгоритм виявився коротшим, всього сім етапів проти восьми. Звісно, у випадку практичного застосування обирається алгоритм, який краще відповідає конкретним вимогам і характеристикам даних, але у подібних, простих задачах, сортування вибором буде приорітетнішим ніж сортування вставками.

Висновок: Основна відмінність між цими двома методами полягає в їхній ефективності та способі роботи. Сортування вибором завжди вибирає найменший (або найбільший) елемент та переміщає його на відповідне місце, що робить його менш ефективним для великих списків. Сортування вставками, з іншого боку, працює добре на малих списках та списках, які вже частково відсортовані.

Отже, обираючи між цими двома методами, слід враховувати характеристики задачі та обсяг даних, які потрібно сортувати, для вибору більш підходящого методу сортування.