**Algorytmy i struktury danych**

**Projekt 3**

***Cel projektu:***

1. Porównaj szybkość działania 4 metod sortowania: Insertion Sort, Selection Sort, Heap Sort, Cocktail Sort dla tablicy liczb całkowitych (rzędu 50k  ‐  200k elementów) generowanych w postaci: losowej, rosnącej, malejącej, stałej, v‐ kształtnej. Przedstaw wykresy t = f(n) dla każdej z metod w zależności od postaci tablicy wejściowej, gdzie: t ‐ czas sortowania; n ‐ liczba elementów tablicy. Liczbę  elementów należy dobrać  w taki sposób, aby możliwe było wykonanie pomiarów. Wyniki przedstawić  na czterech wykresach (przynajmniej 15 punktów pomiarowych).
2. Sformułuj wnioski odnośnie:
   1. złożoności obliczeniowej badanych metod i ich związku z efektywnością sortowania oraz zajętością pamięciową  każdej z nich
   2. wpływu postaci ciągów wejściowych na czas sortowania (najgorsze, najlepsze przypadki).
3. Dla różnych typów danych wejściowych porównaj efektywność działania powyższych algorytmów. Przedstaw wykresy t = f(n) dla każdego typu danych wejściowych i różnych metod sortowania (5 wykresów). Liczbę elementów należy dobrać w taki sposób, aby możliwe było wykonanie pomiarów.
4. Sformułować wnioski odnośnie złożoności obliczeniowej i efektywności wykonywania oraz zachowania się  algorytmów dla poszczególnych typów danych wejściowych.
5. Zaimplementuj algorytm Quicksort w dwóch wersjach: rekurencyjnie oraz iteracyjnie (z własną implementacją  stosu). Porównaj obie wersje na wspólnym wykresie przy sortowaniu ciągu losowego. Następnie porównaj różne sposoby wyboru klucza do porównania: skrajnie prawego, środkowego co do położenia, losowo wybranego. Utwórz wykres porównujący efektywność działania algorytmu (iteracyjnego lub rekurencyjnego) w zależności od wyboru różnego klucza dla A‐kształtnego ciągu wejściowego (przynajmniej 15 punktów pomiarowych).
6. Sformułuj wnioski dotyczące złożoność badanych algorytmów. Jak wybór klucza wpływa na działanie algorytmu (najgorsze, najlepsze przypadki)?
7. **Tabele oraz wykresy [t = f(n)]**

Porównanie czasów dla tablicy **losowej**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Czas [sekundy]** | | | |
| **Ilość elementów** | **Cocktail Sort** | **Heap Sort** | **Insertion Sort** | **Selection Sort** |
| **50000** | 11 | 0,02 | 7,2 | 4,7 |
| **60000** | 15,3 | 0,02 | 10,3 | 6,9 |
| **70000** | 21,6 | 0,03 | 14 | 9,4 |
| **80000** | 28 | 0,04 | 18,5 | 12,3 |
| **90000** | 35,1 | 0,04 | 22,8 | 15,5 |
| **100000** | 43,9 | 0,06 | 29 | 19,2 |
| **110000** | 52,9 | 0,05 | 34,2 | 24,6 |
| **120000** | 63,8 | 0,06 | 40,9 | 28,6 |
| **130000** | 73,8 | 0,06 | 47,9 | 31,4 |
| **140000** | 87,3 | 0,07 | 56,3 | 38,2 |
| **150000** | 100,7 | 0,08 | 67 | 44,9 |
| **160000** | 117,9 | 0,08 | 72,8 | 52,2 |
| **170000** | 131 | 0,09 | 82,8 | 59,8 |
| **180000** | 147,1 | 0,09 | 92,5 | 71,5 |
| **190000** | 165,9 | 0,1 | 108,4 | 82,2 |

Wykres:

Porównanie czasów dla tablicy **rosnącej**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Czas [sekundy]** | | | |
| **Ilość elementów** | **Cocktail Sort** | **Heap Sort** | **Insertion Sort** | **Selection Sort** |
| **50000** | 0,00015 | 0,02 | 5,1 | 5 |
| **60000** | 0,00023 | 0,029 | 6,6 | 7,5 |
| **70000** | 0,00043 | 0,034 | 10,1 | 10,5 |
| **80000** | 0,00045 | 0,036 | 11,7 | 14,4 |
| **90000** | 0,00046 | 0,038 | 14,7 | 17,8 |
| **100000** | 0,00047 | 0,041 | 18,2 | 21 |
| **110000** | 0,00048 | 0,046 | 23,7 | 24,4 |
| **120000** | 0,00051 | 0,054 | 26,2 | 33 |
| **130000** | 0,00057 | 0,06 | 31,3 | 35,1 |
| **140000** | 0,0006 | 0,061 | 35,7 | 37,7 |
| **150000** | 0,00064 | 0,063 | 43,4 | 46,2 |
| **160000** | 0,00072 | 0,069 | 47,1 | 64,3 |
| **170000** | 0,00077 | 0,073 | 56,2 | 72,9 |
| **180000** | 0,001 | 0,078 | 63,45 | 84,1 |
| **190000** | 0,0012 | 0,086 | 67,43 | 86,9 |

Wykresy (aby dokładniej uwidocznić różnice należało rozłożyć metody sortowań na dwa osobne wykresy):

Porównanie czasów dla tablicy **malejącej**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Czas [sekundy]** | | | |
| **Ilośc elementów** | **Cocktail Sort** | **Heap Sort** | **Insertion Sort** | **Selection Sort** |
| **50000** | 18,2 | 0,02 | 9,6 | 7,7 |
| **60000** | 24,8 | 0,02 | 13,6 | 10,9 |
| **70000** | 35,8 | 0,03 | 19 | 14,9 |
| **80000** | 45,6 | 0,03 | 24,5 | 18,7 |
| **90000** | 57,6 | 0,03 | 31,6 | 24,1 |
| **100000** | 69,4 | 0,03 | 38 | 33,5 |
| **110000** | 85,6 | 0,04 | 49,4 | 44,8 |
| **120000** | 123,6 | 0,04 | 58,1 | 54,7 |
| **130000** | 132,33 | 0,05 | 67,1 | 62 |
| **140000** | 137,7 | 0,05 | 78,7 | 69,6 |
| **150000** | 142,3 | 0,06 | 94,8 | 72,1 |
| **160000** | 153,1 | 0,06 | 98,8 | 82,9 |
| **170000** | 172,6 | 0,06 | 112,2 | 95,2 |
| **180000** | 195,2 | 0,07 | 125 | 112,1 |
| **190000** | 250,8 | 0,078 | 139,4 | 125 |

Wykres:

Porównanie czasów dla tablicy **stałej**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Czas [sekundy]** | | | |
| **Ilość elementów** | **Cocktail Sort** | **Heap Sort** | **Insertion Sort** | **Selection Sort** |
| **50000** | 0,00021 | 0,0015 | 4,6 | 4,8 |
| **60000** | 0,00024 | 0,0017 | 6,5 | 8,7 |
| **70000** | 0,00031 | 0,0021 | 8,9 | 9,74 |
| **80000** | 0,00035 | 0,0027 | 11,6 | 12,48 |
| **90000** | 0,00036 | 0,0026 | 14,7 | 15,78 |
| **100000** | 0,0004 | 0,0034 | 18,9 | 22,33 |
| **110000** | 0,00047 | 0,0034 | 22,4 | 26,22 |
| **120000** | 0,00062 | 0,0034 | 26,36 | 27,9 |
| **130000** | 0,00063 | 0,0035 | 32,23 | 34,44 |
| **140000** | 0,00065 | 0,0043 | 38,09 | 39,63 |
| **150000** | 0,00074 | 0,0046 | 42,42 | 42,07 |
| **160000** | 0,00087 | 0,0047 | 50,3 | 48,98 |
| **170000** | 0,00088 | 0,0048 | 60,8 | 55,23 |
| **180000** | 0,0009 | 0,0054 | 67,14 | 61,2 |
| **190000** | 0,00092 | 0,0059 | 76,75 | 69,4 |

Wykresy (aby dokładniej uwidocznić różnice należało rozłożyć metody sortowań na dwa osobne wykresy):

Porównanie czasów dla tablicy **V-kształtnej:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Czas [sekundy]** | | | |
| **Ilość elementów** | **Cocktail Sort** | **Heap Sort** | **Insertion Sort** | **Selection Sort** |
| **50000** | 0,0005 | 0,02 | 4,56 | 4,8 |
| **60000** | 0,001 | 0,025 | 6,54 | 6,6 |
| **70000** | 0,0011 | 0,028 | 8,96 | 9,5 |
| **80000** | 0,0011 | 0,032 | 12,42 | 13,69 |
| **90000** | 0,00123 | 0,037 | 15,09 | 16,23 |
| **100000** | 0,00128 | 0,041 | 19,74 | 19,06 |
| **110000** | 0,0013 | 0,045 | 23,22 | 23,56 |
| **120000** | 0,0014 | 0,059 | 28,41 | 28,66 |
| **130000** | 0,0014 | 0,064 | 34,85 | 33,99 |
| **140000** | 0,0015 | 0,08 | 40,3 | 38,73 |
| **150000** | 0,0017 | 0,088 | 44,66 | 44 |
| **160000** | 0,002 | 0,09 | 53,01 | 54,58 |
| **170000** | 0,0021 | 0,091 | 54,47 | 57,48 |
| **180000** | 0,0025 | 0,092 | 59,95 | 71,39 |
| **190000** | 0,0026 | 0,095 | 67,18 | 76,18 |

Wykresy (aby dokładniej uwidocznić różnice należało rozłożyć metody sortowań na dwa osobne wykresy):

1. **Wnioski:**

**Insertion Sort**

ZŁOŻONOŚĆ: O(n2)

**Zalety:**

- Prosty w implementacji

- Stabilny

- Jeżeli tablica jest posortowana, jest szybki

**Wady:**

- Bardzo wolny dla tablicy losowej oraz v-kształtnej

**Selection Sort**

ZŁOŻONOŚĆ: O(n2)

**Zalety:**

- Prosty w implementacji

**Wady:**

- Niestabilny

- Bardzo wolny dla każdego rodzaju tablicy

**Heap Sort**

ZŁOŻONOŚĆ: O(n log2n)

**Zalety:**

- Niezależnie od rodzaju tablicy jest bardzo szybki

- Przydatny przy bardzo dużych tablicach

**Wady:**

- Trudniejszy w implementacji

**Cocktail Sort**

ZŁOŻONOŚĆ: O(n2)

**Zalety:**

- Podobnie jak insertion sort jest bardzo szybki dla posortowanych tablic

**Wady:**

- Bardzo wolny dla tablicy losowej

1. **Porównanie różnych typów danych (byte, integer, double)**

**Insertion Sort**

Tabele dla każdego rodzaju tablic:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Losowa | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 8,04948 | 7,2 | 9,63981 |
| 60000 | 11,96779 | 10,3 | 14,16651 |
| 70000 | 18,28372 | 14 | 20,41997 |
| 80000 | 19,87506 | 18,5 | 27,81088 |
| 90000 | 24,76773 | 22,8 | 33,02223 |
| 100000 | 29,81994 | 29 | 41,21938 |
| 110000 | 36,5427 | 34,2 | 49,66967 |
| 120000 | 42,94596 | 40,9 | 57,32933 |
| 130000 | 51,6052 | 47,9 | 71,56645 |
| 140000 | 64,14922 | 56,3 | 81,84996 |
| 150000 | 67,33607 | 67 | 92,77803 |
| 160000 | 76,6669 | 72,8 | 104,5244 |
| 170000 | 85,4993 | 82,8 | 119,6599 |
| 180000 | 97,17097 | 92,5 | 130,926 |
| 190000 | 107,3513 | 108,4 | 145,6616 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Rosnąca | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 4,87064 | 5,13 | 5,73064 |
| 60000 | 8,28132 | 6,63 | 8,71092 |
| 70000 | 12,79119 | 10,12 | 11,25558 |
| 80000 | 15,40678 | 11,72 | 14,69479 |
| 90000 | 21,83628 | 14,76 | 18,42147 |
| 100000 | 24,16302 | 18,2 | 24,74696 |
| 110000 | 29,63612 | 23,72 | 35,12533 |
| 120000 | 35,0451 | 26,23 | 43,84127 |
| 130000 | 42,75927 | 31,38 | 46,29514 |
| 140000 | 49,13515 | 35,78 | 48,65506 |
| 150000 | 53,58822 | 43,49 | 53,99142 |
| 160000 | 66,54836 | 47,11 | 73,2153 |
| 170000 | 71,42883 | 56,28 | 75,61378 |
| 180000 | 80,26043 | 63,45 | 87,69757 |
| 190000 | 88,0887 | 70,43 | 94,21997 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Malejąca | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 12,32078 | 9,96 | 10,407 |
| 60000 | 17,99403 | 13,65 | 15,46245 |
| 70000 | 25,07948 | 19 | 22,4771 |
| 80000 | 32,15019 | 24,53 | 29,15082 |
| 90000 | 42,16758 | 31,6 | 40,37886 |
| 100000 | 53,4439 | 38 | 51,18261 |
| 110000 | 58,94852 | 49,46 | 61,51419 |
| 120000 | 73,5648 | 58,1 | 67,4932 |
| 130000 | 83,61904 | 67,19 | 73,16092 |
| 140000 | 88,23873 | 78,75 | 99,34715 |
| 150000 | 95,76255 | 94,88 | 115,9465 |
| 160000 | 112,2211 | 98,84 | 129,6669 |
| 170000 | 122,8988 | 112,26 | 138,3935 |
| 180000 | 139,9337 | 125 | 143,0773 |
| 190000 | 154,1415 | 139,42 | 184,5141 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Stała | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 7,60615 | 4,61 | 5,05438 |
| 60000 | 8,93432 | 6,54 | 7,31916 |
| 70000 | 12,20589 | 8,99 | 9,91055 |
| 80000 | 15,9337 | 11,65 | 12,82224 |
| 90000 | 21,10749 | 14,76 | 17,0517 |
| 100000 | 24,94096 | 18,96 | 20,19073 |
| 110000 | 28,99816 | 22,46 | 25,31095 |
| 120000 | 34,70105 | 26,36 | 29,27555 |
| 130000 | 40,7074 | 32,23 | 34,06214 |
| 140000 | 46,62556 | 38,09 | 40,08496 |
| 150000 | 57,11877 | 42,42 | 46,12461 |
| 160000 | 55,09614 | 50,33 | 52,56628 |
| 170000 | 63,51144 | 57,85 | 59,52632 |
| 180000 | 73,68538 | 67,14 | 66,13757 |
| 190000 | 88,43287 | 76,75 | 74,63658 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | V-kształtna | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 5,1236 | 4,56 | 6,97687 |
| 60000 | 6,93748 | 6,54 | 9,61973 |
| 70000 | 9,81927 | 8,96 | 14,92002 |
| 80000 | 12,42488 | 12,42 | 17,33618 |
| 90000 | 16,44523 | 15,1 | 25,92401 |
| 100000 | 20,74723 | 19,74 | 27,15373 |
| 110000 | 23,46106 | 23,22 | 33,7919 |
| 120000 | 37,12484 | 28,41 | 40,09595 |
| 130000 | 41,02604 | 34,85 | 46,47362 |
| 140000 | 45,27307 | 40,3 | 54,84959 |
| 150000 | 61,4525 | 44,66 | 62,12904 |
| 160000 | 64,97076 | 53,01 | 72,10925 |
| 170000 | 73,94506 | 61,47 | 71,12339 |
| 180000 | 85,74883 | 69,95 | 66,44523 |
| 190000 | 92,12585 | 80,18 | 74,2579 |

Wykresy dla każdego rodzaju tablic:

**Selection Sort**

Tabele dla każdego rodzaju tablic:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Losowa | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 4,66633 | 4,7 | 6,98307 |
| 60000 | 6,72312 | 6,9 | 10,07342 |
| 70000 | 9,12061 | 9,4 | 13,89323 |
| 80000 | 11,83759 | 12,3 | 18,50043 |
| 90000 | 15,12152 | 15,5 | 25,14938 |
| 100000 | 19,18226 | 19,2 | 29,14415 |
| 110000 | 22,37284 | 24,6 | 35,43836 |
| 120000 | 26,85116 | 28,6 | 41,21847 |
| 130000 | 31,21259 | 31,4 | 45,5627 |
| 140000 | 36,56755 | 38,2 | 54,25199 |
| 150000 | 42,45692 | 44,9 | 62,03002 |
| 160000 | 48,30632 | 52,2 | 72,51899 |
| 170000 | 53,5592 | 59,8 | 80,09808 |
| 180000 | 61,20351 | 71,5 | 88,18547 |
| 190000 | 68,29686 | 82,2 | 100,2693 |
|  |  |  |  |
|  |  |  |  |
|  | Rosnąca | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 5,80247 | 5 | 6,52728 |
| 60000 | 8,23447 | 7,5 | 9,68781 |
| 70000 | 11,62564 | 10,5 | 13,20296 |
| 80000 | 13,62538 | 14,4 | 17,04266 |
| 90000 | 18,33289 | 17,8 | 23,01823 |
| 100000 | 21,54707 | 21 | 27,53969 |
| 110000 | 27,62288 | 24,4 | 32,19916 |
| 120000 | 31,46724 | 33 | 39,07706 |
| 130000 | 37,89817 | 35,1 | 46,61641 |
| 140000 | 42,82974 | 37,7 | 54,42362 |
| 150000 | 51,16844 | 46,2 | 61,40309 |
| 160000 | 60,12922 | 64,3 | 69,71874 |
| 170000 | 65,79078 | 72,9 | 79,76956 |
| 180000 | 70,43293 | 84,1 | 89,42874 |
| 190000 | 73,75536 | 86,9 | 97,76773 |
|  |  |  |  |
|  | Malejąca | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 4,66904 | 7,7 | 7,50695 |
| 60000 | 6,74609 | 10,9 | 10,36299 |
| 70000 | 9,11583 | 14,9 | 14,05001 |
| 80000 | 12,02931 | 18,7 | 18,12457 |
| 90000 | 15,05271 | 24,1 | 24,14757 |
| 100000 | 18,64163 | 33,5 | 28,93397 |
| 110000 | 22,79635 | 44,8 | 34,60183 |
| 120000 | 26,8985 | 54,7 | 42,01139 |
| 130000 | 31,63498 | 62 | 49,40382 |
| 140000 | 36,58279 | 69,6 | 58,8155 |
| 150000 | 43,75509 | 72,1 | 69,76687 |
| 160000 | 47,92706 | 82,9 | 80,15841 |
| 170000 | 54,1261 | 95,2 | 92,43878 |
| 180000 | 60,90851 | 112,1 | 99,02315 |
| 190000 | 67,55741 | 125 | 106,6914 |
|  |  |  |  |
|  | Stała | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 4,63026 | 4,8 | 6,52202 |
| 60000 | 6,6847 | 8,7 | 9,3549 |
| 70000 | 9,12985 | 9,74 | 14,96117 |
| 80000 | 13,35475 | 12,48 | 16,60119 |
| 90000 | 15,22143 | 15,78 | 21,11569 |
| 100000 | 18,56219 | 22,33 | 28,10054 |
| 110000 | 23,84371 | 26,22 | 31,59176 |
| 120000 | 27,10537 | 27,9 | 39,46122 |
| 130000 | 32,87948 | 34,44 | 45,96884 |
| 140000 | 36,76197 | 39,63 | 53,3345 |
| 150000 | 42,98651 | 42,07 | 61,93232 |
| 160000 | 48,40525 | 48,98 | 69,05502 |
| 170000 | 54,82817 | 55,23 | 77,34808 |
| 180000 | 61,4767 | 61,2 | 87,50762 |
| 190000 | 68,26506 | 69,4 | 99,65781 |
|  |  |  |  |
|  |  |  |  |
|  | V-kształtna | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 4,59462 | 4,8 | 6,5315 |
| 60000 | 7,76856 | 6,6 | 9,42536 |
| 70000 | 9,24547 | 9,5 | 13,89692 |
| 80000 | 11,79548 | 13,69 | 17,54309 |
| 90000 | 16,82015 | 16,23 | 22,05308 |
| 100000 | 19,54989 | 19,06 | 28,65393 |
| 110000 | 23,1662 | 23,56 | 31,64864 |
| 120000 | 26,74469 | 28,66 | 39,53891 |
| 130000 | 33,55471 | 33,99 | 46,52867 |
| 140000 | 37,41818 | 38,73 | 51,7388 |
| 150000 | 42,48118 | 44 | 61,96363 |
| 160000 | 48,55101 | 54,58 | 69,92007 |
| 170000 | 54,61529 | 57,48 | 78,2618 |
| 180000 | 61,61979 | 71,39 | 86,56352 |
| 190000 | 68,11919 | 76,18 | 100,0637 |

Wykresy dla każdego rodzaju tablic:

**Heap Sort**

Tabele dla każdego rodzaju tablic:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Losowa | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 0,02142 | 0,02 | 0,05333 |
| 60000 | 0,02982 | 0,02 | 0,05289 |
| 70000 | 0,03175 | 0,03 | 0,05427 |
| 80000 | 0,03562 | 0,04 | 0,06172 |
| 90000 | 0,04404 | 0,04 | 0,10373 |
| 100000 | 0,04593 | 0,06 | 0,07867 |
| 110000 | 0,05121 | 0,05 | 0,08751 |
| 120000 | 0,05536 | 0,06 | 0,13282 |
| 130000 | 0,061 | 0,06 | 0,10654 |
| 140000 | 0,06846 | 0,07 | 0,20513 |
| 150000 | 0,07147 | 0,08 | 0,16622 |
| 160000 | 0,07811 | 0,08 | 0,11441 |
| 170000 | 0,08031 | 0,09 | 0,14717 |
| 180000 | 0,08965 | 0,09 | 0,15468 |
| 190000 | 0,08926 | 0,1 | 0,13652 |
|  |  |  |  |
|  |  |  |  |
|  | Rosnąca | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 0,01865 | 0,02 | 0,02789 |
| 60000 | 0,02061 | 0,029 | 0,03301 |
| 70000 | 0,02453 | 0,034 | 0,03803 |
| 80000 | 0,02783 | 0,036 | 0,04589 |
| 90000 | 0,03577 | 0,038 | 0,07717 |
| 100000 | 0,036 | 0,041 | 0,05732 |
| 110000 | 0,0397 | 0,046 | 0,06484 |
| 120000 | 0,04276 | 0,054 | 0,09437 |
| 130000 | 0,04667 | 0,06 | 0,07494 |
| 140000 | 0,05404 | 0,061 | 0,11001 |
| 150000 | 0,05622 | 0,063 | 0,09357 |
| 160000 | 0,06432 | 0,069 | 0,12331 |
| 170000 | 0,06979 | 0,073 | 0,10248 |
| 180000 | 0,08759 | 0,078 | 0,13448 |
| 190000 | 0,08567 | 0,086 | 0,13526 |
|  |  |  |  |
|  | Malejąca | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 0,01789 | 0,02 | 0,02528 |
| 60000 | 0,02361 | 0,02 | 0,03143 |
| 70000 | 0,02697 | 0,03 | 0,03757 |
| 80000 | 0,02894 | 0,03 | 0,04187 |
| 90000 | 0,03303 | 0,03 | 0,07282 |
| 100000 | 0,03903 | 0,03 | 0,0602 |
| 110000 | 0,04126 | 0,04 | 0,06383 |
| 120000 | 0,04698 | 0,04 | 0,08639 |
| 130000 | 0,04917 | 0,05 | 0,07472 |
| 140000 | 0,05628 | 0,05 | 0,09632 |
| 150000 | 0,05709 | 0,06 | 0,09043 |
| 160000 | 0,06367 | 0,06 | 0,11724 |
| 170000 | 0,06855 | 0,06 | 0,10347 |
| 180000 | 0,06995 | 0,07 | 0,13164 |
| 190000 | 0,08043 | 0,078 | 0,11362 |
|  |  |  |  |
|  | Stała | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 0,00147 | 0,0015 | 0,00417 |
| 60000 | 0,00164 | 0,0017 | 0,00403 |
| 70000 | 0,00195 | 0,0021 | 0,0041 |
| 80000 | 0,00233 | 0,0027 | 0,00416 |
| 90000 | 0,00247 | 0,0026 | 0,00418 |
| 100000 | 0,00285 | 0,0034 | 0,00521 |
| 110000 | 0,00303 | 0,0034 | 0,00613 |
| 120000 | 0,00323 | 0,0034 | 0,00635 |
| 130000 | 0,0037 | 0,0035 | 0,0067 |
| 140000 | 0,00487 | 0,0043 | 0,00755 |
| 150000 | 0,00504 | 0,0046 | 0,00795 |
| 160000 | 0,00534 | 0,0047 | 0,0079 |
| 170000 | 0,00512 | 0,0048 | 0,00825 |
| 180000 | 0,005 | 0,0054 | 0,0095 |
| 190000 | 0,00545 | 0,0059 | 0,00991 |
|  |  |  |  |
|  | V-kształtna | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 0,01789 | 0,02 | 0,02953 |
| 60000 | 0,02067 | 0,025 | 0,04633 |
| 70000 | 0,02423 | 0,028 | 0,05054 |
| 80000 | 0,02944 | 0,032 | 0,04767 |
| 90000 | 0,03198 | 0,037 | 0,05551 |
| 100000 | 0,03601 | 0,041 | 0,08575 |
| 110000 | 0,04059 | 0,045 | 0,06538 |
| 120000 | 0,04423 | 0,059 | 0,07392 |
| 130000 | 0,04799 | 0,064 | 0,08 |
| 140000 | 0,0553 | 0,08 | 0,08744 |
| 150000 | 0,05574 | 0,088 | 0,12678 |
| 160000 | 0,06003 | 0,09 | 0,1216 |
| 170000 | 0,06489 | 0,091 | 0,11662 |
| 180000 | 0,07254 | 0,092 | 0,13967 |
| 190000 | 0,0752 | 0,095 | 0,1388 |

Wykresy dla każdego rodzaju tablic:

**Cocktail Sort**

Tabele dla każdego rodzaju tablic:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Losowa | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 14,77965 | 11 | 15,54583 |
| 60000 | 19,62476 | 15,3 | 21,50347 |
| 70000 | 24,78974 | 21,6 | 30,76236 |
| 80000 | 32,26383 | 28 | 45,62111 |
| 90000 | 42,77806 | 35,1 | 55,99088 |
| 100000 | 54,00237 | 43,9 | 68,16862 |
| 110000 | 60,26325 | 52,9 | 81,86473 |
| 120000 | 79,55109 | 63,8 | 87,0639 |
| 130000 | 84,79196 | 73,8 | 106,3919 |
| 140000 | 88,38633 | 87,3 | 126,565 |
| 150000 | 107,7508 | 100,7 | 149,9046 |
| 160000 | 125,7503 | 117,9 | 181,6737 |
| 170000 | 141,3591 | 131 | 169,77 |
| 180000 | 156,5081 | 147,1 | 197,8682 |
| 190000 | 175,4446 | 165,9 | 208,4959 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | Rosnąca | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 0,00056 | 0,00015 | 0,00011 |
| 60000 | 0,0004 | 0,00023 | 0,0004 |
| 70000 | 0,00033 | 0,00043 | 0,00084 |
| 80000 | 0,00036 | 0,00045 | 0,00066 |
| 90000 | 0,00038 | 0,00046 | 0,00101 |
| 100000 | 0,00043 | 0,00047 | 0,00141 |
| 110000 | 0,00072 | 0,00048 | 0,0006 |
| 120000 | 0,00053 | 0,00051 | 0,00094 |
| 130000 | 0,00058 | 0,00057 | 0,00112 |
| 140000 | 0,00063 | 0,0006 | 0,00167 |
| 150000 | 0,00066 | 0,00064 | 0,00146 |
| 160000 | 0,00073 | 0,00072 | 0,0009 |
| 170000 | 0,00073 | 0,00077 | 0,00103 |
| 180000 | 0,00115 | 0,001 | 0,00112 |
| 190000 | 0,00134 | 0,0012 | 0,00224 |
|  |  |  |  |
|  | Malejąca | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 17,76099 | 18,2 | 19,32536 |
| 60000 | 26,80972 | 24,8 | 27,89566 |
| 70000 | 36,31502 | 35,8 | 41,18192 |
| 80000 | 46,88733 | 45,6 | 58,0777 |
| 90000 | 51,20791 | 57,6 | 69,63714 |
| 100000 | 65,23338 | 69,4 | 77,18037 |
| 110000 | 75,58057 | 85,6 | 96,18584 |
| 120000 | 102,9442 | 123,6 | 107,3264 |
| 130000 | 120,6888 | 132,33 | 130,2837 |
| 140000 | 131,5259 | 137,7 | 150,1781 |
| 150000 | 173,4762 | 142,3 | 204,1639 |
| 160000 | 205,1864 | 153,1 | 228,9583 |
| 170000 | 217,1481 | 172,6 | 258,9871 |
| 180000 | 236,4804 | 195,2 | 321,6284 |
| 190000 | 277,0076 | 250,8 | 317,2612 |
|  |  |  |  |
|  | Stała | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 0,00022 | 0,00021 | 0,00036 |
| 60000 | 0,00023 | 0,00024 | 0,00038 |
| 70000 | 0,00031 | 0,00031 | 0,00038 |
| 80000 | 0,00034 | 0,00035 | 0,00051 |
| 90000 | 0,00039 | 0,00036 | 0,00061 |
| 100000 | 0,00061 | 0,0004 | 0,00063 |
| 110000 | 0,00042 | 0,00047 | 0,0006 |
| 120000 | 0,00081 | 0,00062 | 0,00112 |
| 130000 | 0,00052 | 0,00063 | 0,00096 |
| 140000 | 0,0006 | 0,00065 | 0,00097 |
| 150000 | 0,00064 | 0,00074 | 0,00096 |
| 160000 | 0,0007 | 0,00087 | 0,00101 |
| 170000 | 0,00084 | 0,00088 | 0,00126 |
| 180000 | 0,00077 | 0,0009 | 0,00113 |
| 190000 | 0,00121 | 0,00092 | 0,00118 |
|  |  |  |  |
|  | V-kształtna | | |
| Ilość elementów | Byte | Integer | Double |
| 50000 | 0,00038 | 0,0005 | 0,00061 |
| 60000 | 0,00044 | 0,001 | 0,00072 |
| 70000 | 0,00065 | 0,0011 | 0,00168 |
| 80000 | 0,00072 | 0,0011 | 0,00098 |
| 90000 | 0,00081 | 0,00123 | 0,00104 |
| 100000 | 0,00083 | 0,00128 | 0,00173 |
| 110000 | 0,00095 | 0,0013 | 0,00136 |
| 120000 | 0,00105 | 0,0014 | 0,00189 |
| 130000 | 0,00104 | 0,0014 | 0,0016 |
| 140000 | 0,00123 | 0,0015 | 0,00225 |
| 150000 | 0,00128 | 0,0017 | 0,00174 |
| 160000 | 0,00134 | 0,002 | 0,00192 |
| 170000 | 0,00212 | 0,0021 | 0,002 |
| 180000 | 0,00151 | 0,0025 | 0,00203 |
| 190000 | 0,00159 | 0,0026 | 0,00397 |

Wykresy dla każdego rodzaju tablic:

1. **Wnioski**

Najwydajniejszym algorytmem sortowania pomimo bardziej skomplikowanej implementacji okazuje się być **Heap Sort**. Różnice w typie tablicy a także jej rozmiarze mają znikomy wpływ na czas z jakim algorytm ten wykonuje sortowanie. Różnice są niezauważalne.

Najgorszym ale też najprostszym w implementacji algorytmem okazał się **Selection Sort**. Przy prawie każdym typie tablicy zachowuje się najgorzej. Czasy są zauważalne nawet przy bardzo małych tablicach.

1. **Porównanie QuickSort**

Wyniki dla tablicy A-kształtnej przy standardowym wyborze klucza:

|  |  |  |
| --- | --- | --- |
| **Ilość elementów** | **QuickSort (iteracyjnie)** | **QuickSort (rekurencyjnie)** |
| **50000** | 0,00357 | 0,00377 |
| **60000** | 0,00459 | 0,0042 |
| **70000** | 0,00513 | 0,00455 |
| **80000** | 0,0054 | 0,00497 |
| **90000** | 0,00721 | 0,00584 |
| **100000** | 0,00843 | 0,00675 |
| **110000** | 0,00879 | 0,00778 |
| **120000** | 0,00922 | 0,00825 |
| **130000** | 0,01 | 0,00978 |
| **140000** | 0,01169 | 0,01012 |
| **150000** | 0,01263 | 0,01039 |
| **160000** | 0,01278 | 0,01106 |
| **170000** | 0,01301 | 0,01233 |
| **180000** | 0,01412 | 0,01383 |
| **190000** | 0,01501 | 0,01476 |

Porównanie szybkości przez wzgląd na wybór klucza:

|  |  |  |  |
| --- | --- | --- | --- |
| **Ilość elementów** | **Losowy** | **Skrajnie prawy** | **Środkowy** |
| **50000** | 0,016215851 | 0,028864215 | 0,013459156 |
| **60000** | 0,019151882 | 0,03409035 | 0,015896062 |
| **70000** | 0,027262223 | 0,048526757 | 0,022627645 |
| **80000** | 0,030214742 | 0,05378224 | 0,025078236 |
| **90000** | 0,031658948 | 0,056352928 | 0,026276927 |
| **100000** | 0,037122326 | 0,06607774 | 0,03081153 |
| **110000** | 0,039141781 | 0,069672371 | 0,032487679 |
| **120000** | 0,044624845 | 0,079432225 | 0,037038622 |
| **130000** | 0,051515189 | 0,091697037 | 0,042757607 |
| **140000** | 0,052151812 | 0,092830224 | 0,043286004 |
| **150000** | 0,055251842 | 0,098348278 | 0,045859029 |
| **160000** | 0,05925193 | 0,105468435 | 0,049179102 |
| **170000** | 0,062147448 | 0,110622458 | 0,051582382 |
| **180000** | 0,069151223 | 0,123089177 | 0,057395515 |
| **190000** | 0,070151818 | 0,124870236 | 0,058226009 |
| **200000** | 0,072151818 | 0,128430237 | 0,059886009 |

1. **Wnioski**

Wydajniejszą wersją algorytmu *Quick Sort* jest wersja **rekurencyjna.** Sortowania są czynnością typowo rekurencyjną, więc nie dziwi fakt, że ta wersja jest szybsza. Dodatkowo nie potrzebuje dodatkowej pamięci na stos.

Wybór klucza okazuje się być ważną sprawą. W próbce A-kształtnej, wybór klucza skrajnie prawego prowadzi do najdłuższego czasu sortowania, ze względu na podział tablicy na bardzo dużą część i stosunkowo malutką część, ponieważ wybierana jest najmniejsza/największa wartość. Wybór klucza losowego przyspiesza procedurę, ale wiąże się z tym, że natrafić możemy zarówno na najgorszy jak i na najlepszy przypadek. Wybór klucza środkowego początkowo dzieli tablicę na część posortowaną i nieposortowaną, i z tego miejsca zauważalny jest już wzrost wydajności.