

Algorytmy Lista 2

Działanie programu i sposób uruchamiania:

- zostały zrealizowane wszystkie zadania, jako algorytm hybrydowy wybrałem Dual Pivot Quick Sort z Insertion Sort dla przedziałów mniejszych, domyślnie są to przedziały mniejsze lub równe 5, momentem uruchamiania Insertion Sort decyduje stała LIMIT jest podawana jako parametr funkcji
- wykresy oraz dane do wykresów znajdują się w folderach ex2_test oraz ex3_test
- w pliku sort.py znajdują się wszystkie implementacje algorytmów oraz niezbędnych narzędzi, program odpalamy poprzez plik main.py, w przypadku gdy chcemy zobaczyć instrukcje obsługi programu wystarczy uruchomić program z parametrem „-h”: `python3 main.py -h`, w pliku tester.py znajduje się algorytm którego używałem do wykonywania testów i generowania danych do wykresów
- program sortujący jest bardzo uniwersalny, dzięki użyciu funkcji `eval()` przekształcającej stringa na kod pythona możemy przy wywołaniu programu zdefiniować funkcję `compare` która służy porównywaniu obiektów, domyślnie traktuje tablice wejściową jako liczby zmiennopozycyjne lecz przy odrobinie sprytu możemy np.:
 - sortować pojedyncze literki w tym przypadku rosnąco, wywołanie:
`python3 main.py --type quick --comp 'ord(x) <= ord(y)'`
 - lub pliki ze względu na ilość występujących w nich znaków, wywołanie:
`python3 main.py --type merge --comp 'len(open(x).read()) < len(open(y).read())'`
- definicja funkcji `comapare` jest zawarta w cudzysłów jako argument parametru „--comp”.

Odpowiedzi na pytania:

- stała dla Dual Pivot Quick Sort stojąca przy $n \cdot \ln(n)$ dla liczby porównań to w przybliżeniu 0.8549 natomiast dla zwykłego QuickSort to około 1.554
- warto zmieniać moment przełączania się między różnymi algorytmami w algorytmie hybrydowym, obrazują to wykresy w folderze ex3_test gdzie dla limitu równego 5 (`time_limit5.pdf`) algorytm hybrydowy jest szybszy od pozostałych natomiast, natomiast już dla limitu równego 10 (`time_limit10.pdf`) ta różnica się zaciera