

Kompilacja programu:

```
g++ -o generator generatorTest.cpp
g++ -o zestaw2 zestaw2.cpp
```

Uruchomienie:

```
./generator > input.dat
./zestaw2 < input.dat /* wynik pojawia się na ekranie */
./zestaw2 < input.dat > output.dat /* wynik przekierowany do pliku output.dat */
```

Merge Sort

Szybki algorytm sortujący w czasie $N \log N$.

Algorytm składa się z 2 operacji:

- 1) dzielenie tablicy elementów na dwie połówki i sortowanie ich rekurencyjnie
- 2) scalanie posortowanych fragmentów tablic

Algorytm scalania korzysta z pomocniczej tablicy w której umieszcza posortowane elementy.

Algorytm merge sort jest przykładem paradygmatu divide-and-conquer.

(Bardziej szczegółowe opisy poszczególnych etapów wykonania algorytmu w komentarzach w kodzie źródłowym)

Czasy poszczególnych operacji dla Merge Sort:

Czas sortowania zbioru posortowanego (1000 elementów): 0
Czas sortowania zbioru posortowanego odwrotnie (1000 elementów): 0.144
Czas sortowania zbioru posortowanego z losowym elementem na początku (1000 elementów): 0.137
Czas sortowania zbioru posortowanego z losowym elementem na końcu (1000 elementów): 0.132
Czas sortowania zbioru z losowym rozkładem elementów (1000 elementów): 0.191
Czas sortowania zbioru posortowanego (2000 elementów): 0.194
Czas sortowania zbioru posortowanego odwrotnie (2000 elementów): 0.203
Czas sortowania zbioru posortowanego z losowym elementem na początku (2000 elementów): 0.194
Czas sortowania zbioru posortowanego z losowym elementem na końcu (2000 elementów): 0.2
Czas sortowania zbioru z losowym rozkładem elementów (2000 elementów): 0.456
Czas sortowania zbioru posortowanego (4000 elementów): 0.433
Czas sortowania zbioru posortowanego odwrotnie (4000 elementów): 0.431
Czas sortowania zbioru posortowanego z losowym elementem na początku (4000 elementów): 0.401
Czas sortowania zbioru posortowanego z losowym elementem na końcu (4000 elementów): 0.414
Czas sortowania zbioru z losowym rozkładem elementów (4000 elementów): 0.727
Czas sortowania zbioru posortowanego (8000 elementów): 0.936
Czas sortowania zbioru posortowanego odwrotnie (8000 elementów): 0.935
Czas sortowania zbioru posortowanego z losowym elementem na początku (8000 elementów): 0.856
Czas sortowania zbioru posortowanego z losowym elementem na końcu (8000 elementów): 0.822
Czas sortowania zbioru z losowym rozkładem elementów (8000 elementów): 1.951
Czas sortowania zbioru posortowanego (16000 elementów): 1.764
Czas sortowania zbioru posortowanego odwrotnie (16000 elementów): 1.955
Czas sortowania zbioru posortowanego z losowym elementem na początku (16000 elementów): 2.407
Czas sortowania zbioru posortowanego z losowym elementem na końcu (16000 elementów): 1.769
Czas sortowania zbioru z losowym rozkładem elementów (16000 elementów): 3.549
Czas sortowania zbioru posortowanego (32000 elementów): 5.177
Czas sortowania zbioru posortowanego odwrotnie (32000 elementów): 4.12
Czas sortowania zbioru posortowanego z losowym elementem na początku (32000 elementów): 4.068
Czas sortowania zbioru posortowanego z losowym elementem na końcu (32000 elementów): 3.74
Czas sortowania zbioru z losowym rozkładem elementów (32000 elementów): 6.829
Czas sortowania zbioru posortowanego (64000 elementów): 10.491
Czas sortowania zbioru posortowanego odwrotnie (64000 elementów): 8.09
Czas sortowania zbioru posortowanego z losowym elementem na początku (64000 elementów): 8.11
Czas sortowania zbioru posortowanego z losowym elementem na końcu (64000 elementów): 7.965
Czas sortowania zbioru z losowym rozkładem elementów (64000 elementów): 13.797
Czas sortowania zbioru posortowanego (128000 elementów): 16.814
Czas sortowania zbioru posortowanego odwrotnie (128000 elementów): 19.994
Czas sortowania zbioru posortowanego z losowym elementem na początku (128000 elementów): 19.045
Czas sortowania zbioru posortowanego z losowym elementem na końcu (128000 elementów): 17.677

Czas sortowania zbioru z losowym rozkładem elementów (128000 elementów): 29.897

Czasy poszczególnych operacji dla Quick Sort:

Czas sortowania zbioru posortowanego (1000 elementów): 0.036
Czas sortowania zbioru posortowanego odwrotnie (1000 elementów): 0.033
Czas sortowania zbioru posortowanego z losowym elementem na początku (1000 elementów): 0.04
Czas sortowania zbioru posortowanego z losowym elementem na końcu (1000 elementów): 0.041
Czas sortowania zbioru z losowym rozkładem elementów (1000 elementów): 0.128
Czas sortowania zbioru posortowanego (2000 elementów): 0.092
Czas sortowania zbioru posortowanego odwrotnie (2000 elementów): 0.059
Czas sortowania zbioru posortowanego z losowym elementem na początku (2000 elementów): 0.057
Czas sortowania zbioru posortowanego z losowym elementem na końcu (2000 elementów): 0.088
Czas sortowania zbioru z losowym rozkładem elementów (2000 elementów): 0.227
Czas sortowania zbioru posortowanego (4000 elementów): 0.152
Czas sortowania zbioru posortowanego odwrotnie (4000 elementów): 0.12
Czas sortowania zbioru posortowanego z losowym elementem na początku (4000 elementów): 0.121
Czas sortowania zbioru posortowanego z losowym elementem na końcu (4000 elementów): 0.146
Czas sortowania zbioru z losowym rozkładem elementów (4000 elementów): 0.433
Czas sortowania zbioru posortowanego (8000 elementów): 0.289
Czas sortowania zbioru posortowanego odwrotnie (8000 elementów): 0.237
Czas sortowania zbioru posortowanego z losowym elementem na początku (8000 elementów): 0.255
Czas sortowania zbioru posortowanego z losowym elementem na końcu (8000 elementów): 0.404
Czas sortowania zbioru z losowym rozkładem elementów (8000 elementów): 0.874
Czas sortowania zbioru posortowanego (16000 elementów): 0.61
Czas sortowania zbioru posortowanego odwrotnie (16000 elementów): 0.507
Czas sortowania zbioru posortowanego z losowym elementem na początku (16000 elementów): 0.502
Czas sortowania zbioru posortowanego z losowym elementem na końcu (16000 elementów): 0.616
Czas sortowania zbioru z losowym rozkładem elementów (16000 elementów): 2.14
Czas sortowania zbioru posortowanego (32000 elementów): 1.465
Czas sortowania zbioru posortowanego odwrotnie (32000 elementów): 1.445
Czas sortowania zbioru posortowanego z losowym elementem na początku (32000 elementów): 1.247
Czas sortowania zbioru posortowanego z losowym elementem na końcu (32000 elementów): 1.404
Czas sortowania zbioru z losowym rozkładem elementów (32000 elementów): 3.257
Czas sortowania zbioru posortowanego (64000 elementów): 2.871
Czas sortowania zbioru posortowanego odwrotnie (64000 elementów): 2.384
Czas sortowania zbioru posortowanego z losowym elementem na początku (64000 elementów): 2.245
Czas sortowania zbioru posortowanego z losowym elementem na końcu (64000 elementów): 2.757
Czas sortowania zbioru z losowym rozkładem elementów (64000 elementów): 6.512
Czas sortowania zbioru posortowanego (128000 elementów): 6.255
Czas sortowania zbioru posortowanego odwrotnie (128000 elementów): 4.852
Czas sortowania zbioru posortowanego z losowym elementem na początku (128000 elementów): 4.793
Czas sortowania zbioru posortowanego z losowym elementem na końcu (128000 elementów): 5.775
Czas sortowania zbioru z losowym rozkładem elementów (128000 elementów): 13.374