

Zadanie 3: zmodyfikuj programy sortujące z zadania pierwszego tak, aby zliczały ile było porównań podczas wykonania programu. Porównanie to każde wyrażenie typu $x < y$, $x > y$, $x == y$ itd. Na przykład w takim algorytmie wykonujemy $n-1$ porównań, zlicza je zmienna „ile”:

Require: tablica A o rozmiarze n $\{A=[0, \dots, n-1]\}$

Algorytm: MAX

```
1: max = 0;
2: ile = 0;
3: for all i = 1 to n do
4:     ile = ile + 1;
5:     if A[i] > A[max] then
6:         max = i ;
7:     end if
8:     i = i + 1;
9: end for
10: return max, ile;
```

(nie trzeba tego algorytmu implementować ani nic z nim robić, jest tylko po to aby pokazać jak liczymy porównania)

Wyślij na moodle kody źródłowe zmodyfikowanych programów oraz liczby porównań zliczone podczas sortowania plików testowych (rand.txt, alfa.txt, back.txt, uniq.txt). Liczby porównań najlepiej przesłać w formie wykresu, gdzie liczba porównań pokazana jest w funkcji n (od 1000 do 1000000) dla każdego z 5 programów sortujących. Jeśli nie uda się zrobić wykresu, za podanie liczby porównań w formie tekstu, tabeli albo arkusza kalkulacyjnego będzie 75% punktów. Zakładając że wszystkie 5 programów uda się przedstawić na 1 wykresie, wykresów powinno być 4 (po jednym dla danych rand.txt, alfa.txt, back.txt, uniq.txt).

Dodatkowe: oprócz liczby zliczeń policz także czas wykonywania.