

Algorytmy i struktury danych

Laboratorium 3

Termin oddania: 15 kwiecień, 10:34

Zadanie 1. [30%]

Uzupełnij **Zadanie 2.** z Listy 2 o algorytm RADIX SORT. Porównaj wcześniej zaimplementowane algorytmy z RADIX SORTEM pod względem liczby wykonanych operacji, czasu działania algorytmu oraz zużycia pamięci (wykorzystaj np. `sysinfo` lub `GetProcessMemoryInfo`), dla danych o rozmiarze $n \in \{10, 50, 100, 500, 1000, 5000, \dots, 100000\}$, sprawdź jak na działanie RADIX SORTa wpływa zakres, z którego losowane są dane do sortowania.

Zadanie 2. [55%]

Zaimplementuj podane na wykładzie algorytmy: RANDOMIZED SELECT, SELECT. Program przyjmuje jeden z dwóch parametrów wejściowych:

- wywołanie `./main -r` oznacza operowanie na danych losowych długości n ,
- wywołanie `./main -p` oznacza operowanie na losowej permutacji zbioru $\{1, 2, \dots, n\}$.

Po uruchomieniu, program wczytuje ze standardowego wejścia dwie liczby całkowite: n —długość danych oraz $1 \leq k \leq n$ —numeru szukanej statystyki pozycyjnej, a następnie generuje tablicę danych (zależnie od parametru uruchomienia) i sekwencyjnie uruchamia zaimplementowane algorytmy na wygenerowanych danych. W czasie wykonywania algorytmów SELECT, RANDOMIZED SELECT, na standardowym wyjściu błędów, powinien być wypisywany log, tak by można było działanie algorytmu odtworzyć. W szczególności powinien zawierać on tablicę danych, k , kolejno wybierane pivoty, wykonywane porównania i przestawienia oraz podsumowanie zawierające liczbę porównań oraz przestawień elementów (**Uwaga:** w przypadkach, gdy nie jest to niezbędne dla znalezienia k -tej statystyki pozycyjnej, nie powinno następować sortowanie całej tablicy).

Wynikiem, wyświetlanym na standardowym wyjściu, działania algorytmu jest tablica z zaznaczoną k -tą statystyką pozycyjną.

Przykładowe wywołanie:

```
./main -p
7
3
2 1 [3] 5 4 6 7
```

Wykonaj testy, powtarzając wywołania algorytmów dla tych samych danych wejściowych, by wyciągnąć wnioski na temat minimalnej i maksymalnej liczby porównań dla obu algorytmów, policz również średnią i odchylenie standardowe dla zebranych statystyk.

Zadanie 3. [15%]

Zaimplementuj rekurencyjny algorytm wyszukiwania binarnego. Program na wejściu otrzymuje posortowaną tablicę długości n oraz wartość v , i zwraca 1 w przypadku istnienia elementu v w tablicy lub 0 w przeciwnym przypadku. Przetestuj działanie Master theorem dla zliczonej w trakcie działania algorytmu liczby porównań elementów oraz czasu wykonania. Test powtórz dla $n \in \{1000, 2000, \dots, 100000\}$ i na ich podstawie oszacuj czynnik $O(1)$ dla obu tych statystyk.

Zadanie 4. [20% (dodatkowe)]

Wykorzystaj część lub całość algorytmu SELECT w algorytmie QuickSort oraz Dual Pivot QuickSort. Wykonaj testy dla różnych rozmiarów danych i wyciągnij wnioski z porównania czasu wykonania oraz liczby porównań dla algorytmów z Listy 2 z ich odpowiednikami z wykorzystanym algorytmem SELECT.