Algorytmy i struktury danych

Laboratorium - lista 3

Termin wysłania: 2022-05-01

Zadanie 1. [40 p.]
Zaimplementuj podane na wykładzie algorytmy

- RANDOMIZED SELECT oraz
- SELECT.

Po uruchomieniu, program wczytuje ze standardowego wejścia:

- *n* – długość danych wejściowych,
- *k*, $1 \leq k \leq n$, – numer szukanej statystyki pozycyjnej,
- ciąg kluczy (liczb całkowitych z przedziału: $[0, 2n - 1]$).

Podobnie jak w zadaniach z listy 2. zaimplementuj generator danych wejściowych i drukuj stany pośrednie obliczeń oraz *wynik* dla odpowiednio małych danych (nie większych niż 50), a dla wszystkich rozmiarów danych - liczby wykonanych porównań i przestawień.
Przy drukowaniu *wyniku* wyświetl:

- końcowy stan tablicy,
- początkowy stan tablicy,
- znalezioną statystykę pozycyjną
- posortowany ciąg kluczy (aby sprawdzić czy na k-tej pozycji jest znaleziona wartość).

Zadanie 2. [20 p.]
Podobnie jak w zadaniach z listy drugiej, przeprowadź testy zaimplementowanych algorytmów dla kilku różnych wartości parametru *k* mające na celu zbadanie liczby wykonywanych porównań oraz przestawień elementów i przygotuj wykresy ilustrujące wyniki.
W tym celu powtarzaj wywołania algorytmów dla tych samych danych wejściowych.
Testy wykonaj dla $n \in \{100, 200, \ldots, 10\,000\}$.
Dla każdego *n* wykonaj *m* niezależnych powtórzeń, przyjmując np. *m* = 100.

Zadanie 3. [20 p.]
W poznanej na wykładzie i zaimplementowanej w Zadaniu 1. wersji algorytmu SELECT dzielimy tablicę wejściową na *n*/5 grup po 5 elementów każda (ostatnia grupa może mieć mniej elementów).
Zbadaj eksperymentalnie, jaki wpływ na złożoność algorytmu (liczbę porównań, liczbę przestawień elementów oraz czas działania) ma liczba grup, na którą dzielona jest tablica w kroku wyszukiwania mediany median.
W tym celu wykonaj podobne eksperymenty jak w Zadaniu 2. i wyznacz podobne statystyki dla wariantów algorytmu SELECT , w których w etapie wyznaczania mediany median tablica dzielona jest na *n*/*k* grup dla $k \in \{3, 5, 7, 9\}$.
Uzyskane wyniki przedstaw przy pomocy odpowiednich wykresów.
Wyciągnij wnioski z przeprowadzonych eksperymentów.

Zadanie 4. [10 p.]
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 oraz 0 w przeciwnym przypadku.
Zademonstruj poprawność na małych danych.
Przeprowadź testy dla rozmiarów tablicy $n \in \{1000, 2000, \ldots, 100\,000\}$.
Porównaj oszacowania uzyskane z Master Theorem do zliczonej w trakcie działania algorytmu liczby porównań elementów oraz czasu wykonania dla różnych wartości *v* (np. dla elementów „blisko początku", „około środka" i „blisko końca" tablicy) oraz dla elementu spoza tablicy.
Wykonaj także podobny eksperyment, gdzie szukany element wybierany jest losowo spośród elementów znajdujących się w tablicy (dla każdego *n* wykonaj odpowiednią liczbę powtórzeń oraz uśrednij otrzymane wyniki).
Na podstawie uzyskanych wyników oszacuj czynnik O(1) dla obu tych statystyk (liczba porównań i czas wykonania) w każdym przypadku.
Przygotuj odpowiednie wykresy.

Zadanie 5. [10 p.]
Wykorzystaj część lub całość algorytmu SELECT w algorytmie QUICKSORT oraz DUAL PIVOT QUICKSORT .
Zademonstruj działanie i poprawność dla małych danych.
Wykonaj testy dla różnych rozmiarów danych (podobne do tych z Zadania 2. z Listy 2).
Wyciągnij wnioski z porównania średniego czasu wykonania oraz średniej liczby porównań dla algorytmów z Listy 2 z ich odpowiednikami wykorzystującymi algorytm SELECT .
Zbadaj czas działania i liczbę porównań dla „worst-case data" dla bazowej wersji algorytmu QUICKSORT , dla danych, dla których „zwykły" QUICKSORT osiąga pesymistyczną złożoność rzędu $\Theta(n^2)$ (podobnie jak poprzednio, wykonaj testy dla n $\in$ {100, 200, . . . , 10 000}). Wyciągnij wnioski z przeprowadzonych eksperymentów.

W każdym przypadku postaraj się eksperymentalnie wyestymować stałe stojące przy dominującym składniku w wyrażeniu opisującym asymptotyczną liczbę porównań wykonywanych przez badane algorytmy.