

Roznice w czasie realizacji sortowania metodą
quicksort, w zależności od wyboru elementu
rozdzielającego

Arkadiusz Cyktor 200367

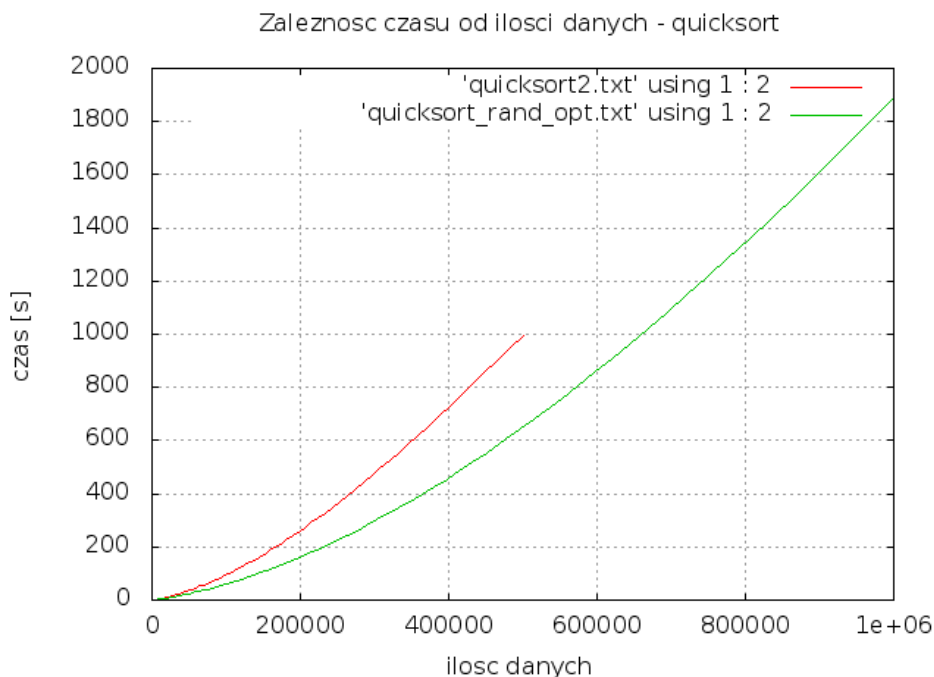
20 kwietnia 2014

Przypadek optymistyczny dla algorytmu *quicksort*, to sytuacja, w której do podziału tablicy za każdym razem uda nam się wybrać medianę sortowanego fragmentu.

Przypadek przeciętny dla algorytmu *quicksort*, to sytuacja, w której występuje równomierne prawdopodobieństwo wyboru elementu dzielącego.

Przypadek pesymistyczny dla algorytmu *quicksort*, to sytuacja, w której sortujemy uporządkowaną już tablicę.

1. Poniższy wykres przedstawia zależność czasu potrzebnego na wykonanie algorytmu **quicksort** dla przypadku przeciętnego. Jak widac jest to funkcja $O(n \log n)$.

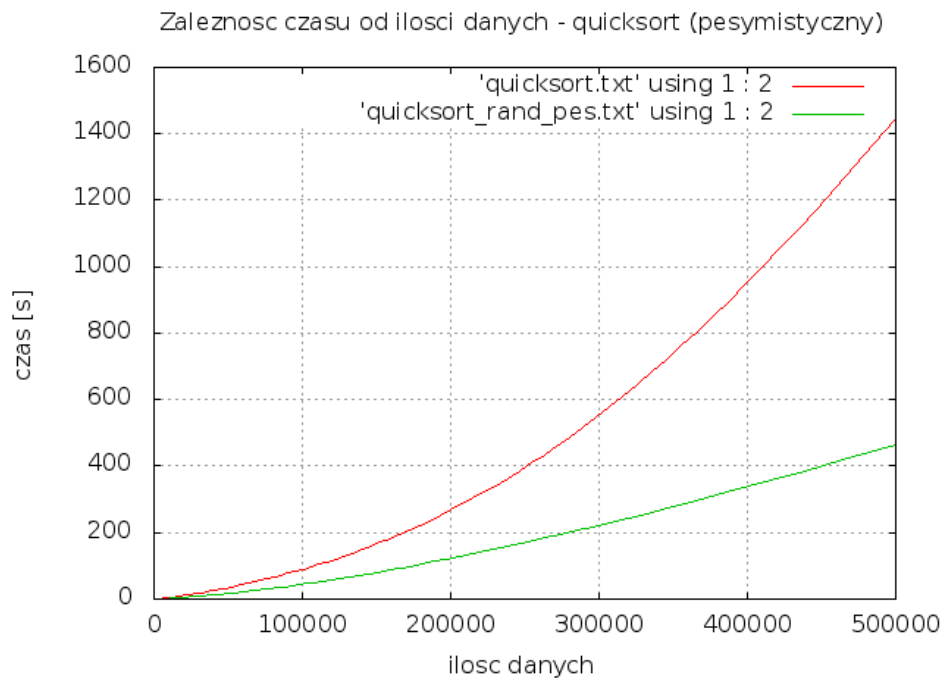


Linia czerwona odpowiada algorytmowi, w którym elementem rozdzielającym zawsze było środkowe pole wektora. Zielona natomiast, obrazuje zależność dla przypadku, gdy element ten wybierany był losowo.

Ze względu na długi czas symulacji, pierwszy z wykresów zatrzymuje się na wartości 500000 elementów, jednak nawet w takiej skali widać, że w obu przypadkach złożoność obliczeniowa rośnie w taki sam sposób, drugie z rozwiązań wydaje się być jednak wydajniejsze. Ponieważ w przypadku optymistycznym dla algorytmu quicksort, złożoność obliczeniowa również zmienia się według funkcji $n \log n$ (Według literatury jest jednak niższa o około 39 procent - dokładana złożoność dla przypadku przeciętnego wynosi $O(2n \ln(n))$, co w przybliżeniu wynosi $O(1,39n \log n)$ ¹⁾, wykresy dla niej prezentowałyby się prawie identycznie jak powyższe (co wiąże się z takimi samymi wnioskami), dlatego postanowiłem ich nie zamieszczać.

1) Obliczenia oparte na informacjach ze strony www.toyoteczka.mainrc.com/pliki/asd/quicksort.pdf

2. Poniższy wykres przedstawia zależność czasu potrzebnego na wykonanie algorytmu od ilości elementów dla przypadku pesymistycznego.



Zielona linia prezentuje zachowanie algorytmu, w którym element rozdzielający wybierany był losowo, natomiast czerwona ten, w którym wybierano element środkowy. Jak widać, dla pierwszego z nich, złożoność jest bardzo zbliżona do tej z przypadku przeciętnego - również rośnie według funkcji $O(n \log n)$. W przypadku drugiego możemy zaobserwować większą zmianę - tutaj złożoność rośnie wykładniczo. Łatwo można dojść więc do wniosku, że losowanie elementu rozdzielającego wpływa bardzo korzystnie na złożoność obliczeniową naszego algorytmu oraz zabezpiecza go przed przypadkiem pesymistycznym.