

## Zadanie 4

### Analiza złożoności QuickSort

Zapoznaj się z opisem oszacowania efektywności algorytmu szybkiego sortowania.

Poniższy fragment pochodzi z książki Davida Gelerntera *Mechaniczne piękno*:

„Przypuśćmy, że leży przed nami stos ponumerowanych kartek, które mamy posortować. (...) Bierzemy ze stosu pierwszą kartkę. Powiedzmy, że ma ona numer 117. Następnie przeglądamy pozostałe kartki. Jeśli numer kolejnej jest mniejszy niż 117, kładziemy ją po lewej stronie pierwszej kartki, jeśli większy – po prawej.

Teraz nasz stos jest podzielony na trzy grupy kartek: te z numerami mniejszymi od 117, kartkę z numerem 117 i kartki z numerami większymi.

Powtarzamy całą operację dla dwóch mniejszych stosów, lewego i prawego.

Stara lewa grupa zostaje w ten sposób podzielona znowu na trzy grupy; to samo dotyczy starej prawej grupy.

Znowu powtarzamy operacje dla każdej nowo powstałej grupy i tak dalej, aż każda grupa będzie się składała z pojedynczej kartki. (...)

Spróbujmy teraz dokonać oszacowania efektywności (...).

Usuwamy pierwszą kartkę i dzielimy pozostałe na dwie grupy: „mniejszą” i „większą”. Jeśli pierwsza kartka miała przeciętny numer, to te dwie grupy będą miały po mniej więcej tyle samo kartek – około połowy kartek ze stosu będzie miało numer mniejszy od numeru pierwszej, a połowa będzie miała numer większy. Następnie powtarzamy procedurę dla kolejnych podstosów i tak dalej. Przypuśćmy, że jakimś szczęśliwym trafem za każdym razem znajdujemy na szczycie kolejnej grupy przeciętną kartkę. Ile razy musielibyśmy przeglądać rosnący szereg grup?

Za pierwszym razem podzieliliśmy stos na dwie grupy, po około  $\frac{1}{2}$  n kartek. Za drugim razem otrzymaliśmy cztery grupy po około  $\frac{1}{4}$  n kartek, za trzecim – osiem grup po  $\frac{1}{8}$  n i tak dalej. Za którym razem otrzymamy szereg grup liczących tylko jedną kartkę każda? Odpowiedź brzmi: logarytm o podstawie 2 z n (...). Powiedzmy, że mamy stos 16 kartek; pierwszy podział daje dwie grupy po 8, drugi – cztery grupy po 4, trzeci – osiem grup po 2, a czwarty – 16 grup złożonych z jednej kartki. Potrzebowaliśmy czterech podziałów: 2 do potęgi 4 daje 16. (...)

Za każdym razem, gdy dokonywaliśmy kolejnego podziału stosu (na dwie, cztery, osiem grup – i tak dalej), musieliśmy sprawdzać [prawie] wszystkie n kartki. Quicksort jest zatem algorytmem o złożoności  $n \cdot \log n$ . (...)

Dla miliona kartek QuikSort potrzebuje około 20 milionów operacji, gdy sortowanie liniowe [przez wstawianie liniowe] – biliona. Załóżmy teraz, że nasz komputer potrafi wykonywać 10 milionów takich operacji na sekundę. Na posortowanie listy zawierającej milion pozycji QuickSort potrzebuje około 2 sekund, a sortowanie liniowe [przez wstawianie liniowe] – czasu rzędu 100 tysięcy sekund, czyli mniej więcej całej doby”.

a) Wykonaj eksperyment ze stosem kart.

b) Wyjaśnij sens wtrącenia „[prawie]”, jakie dodano w przedostatnim akapicie cytatu.