## Liczba operacji

**model liniowej liczby operacji ★** $T = an + b$

model kwadratowej liczby operacji

$$T = an^2 + bn + c$$

Model T(n) = an + b ma wzrost liniowy, czyli O(n). Model T(n) = $an^2$ + bn + c ma wzrost kwadratowy, czyli O($n^2$). W pierwszym modelu zakładamy a > 0, w drugim a > 0. Przy zerowym współczynniku wiodącym stopień wzrostu może być niższy. Notacja O jest ograniczeniem górnym; dla tych modeli z dodatnim współczynnikiem wiodącym dokładniejszą klasą jest odpowiednio Θ(n) i Θ($n^2$). To modele liczby operacji, a nie uniwersalne wzory na czas dowolnego algorytmu.

## Obliczenia równoległe

prawo Amdahla $$S = \frac{1}{(1-p) + \frac{p}{s}}$$

p jest częścią programu, którą można zrównoleglić, a s przyspieszeniem tej części. Wzór zakłada niezmienioną, szeregową część programu; gdy czas części równoległej dąży do zera, niezmieniona część szeregowa ogranicza przyspieszenie do 1/(1 − p), jeśli p < 1. Gdy p = 1, S = s i nie ma skończonej granicy przy s rosnącym bez ograniczeń. Przyjmujemy 0 ≤ p ≤ 1 i s ≥ 1, bez narzutu komunikacji.

## Algorytmy liczbowe

schemat Hornera dla wielomianu

$$b_n = a_n, \qquad b_k = a_k + x b_{k+1}, \qquad W(x) = b_0$$

$W(x) = a_n x^n + \ldots + a_0$; iteracja k = n-1, ..., 0. Kolejność współczynników jest od najwyższej potęgi.

krok algorytmu Euklidesa

$$\gcd(a, b) = \gcd(b, a \bmod b)$$

a,b nieujemne całkowite, b > 0; zakończenie gcd(a,0) = a. gcd oznacza największy wspólny dzielnik (NWD), a mod to reszta z dzielenia całkowitego.

**Algorytm Euklidesa: kolejne reszty**

| krok | a | b | a mod b |
|---|---|---|---|
| 1 | 252 | 105 | 42 |
| 2 | 105 | 42 | 21 |
| 3 | 42 | 21 | 0 |
| 4 | 21 | 0 | — |

NWD(252, 105) = 21.

**Algorytm Euklidesa: kolejne reszty.** W każdym kroku zamień (a, b) na (b, a mod b). Zatrzymaj się przy b = 0: ostatnie niezerowe a jest NWD. Przykład domyślny: NWD(252, 105) = 21; nie wykonuj dzielenia przez zero.

najmniejsza wspólna wielokrotność

$$\mathrm{NWW}(a,b) = \frac{ab}{\gcd(a,b)}$$

Dodatnie liczby całkowite a,b. Dla argumentu zero przyjmujemy NWW = 0 bez dzielenia przez gcd(0,0).

rekurencja ciągu Fibonacciego

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2}$$

n ≥ 2 całkowite; jawnie przyjęte indeksowanie od zera.

## Złożoność i grafy

notacja O — ograniczenie asymptotyczne

$$f(n) \le c g(n) \quad \text{dla } n \ge n_0$$

Istnieją c>0 i $n_0$; funkcje nieujemne od pewnego miejsca. O nie jest dokładnym czasem w sekundach.

liczba podziałów wyszukiwania binarnego

$$k = \lceil \log_2(n+1) \rceil$$

Maksymalna liczba iteracji standardowego wyszukiwania w posortowanej tablicy długości n ≥ 0; sposób liczenia porównań może różnić się od iteracji.

rekurencja sortowania przez scalanie

$$T(n) = 2T(n/2) + \Theta(n) = \Theta(n \log n)$$

Dla n będącego potęgą 2 zapis jest bez zaokrągleń; koszt scalania liniowy.

**Wyszukiwanie binarne i sortowanie przez scalanie**

| krok | przedział / lewa część | element / prawa część | decyzja / wynik |
|---|---|---|---|
| 1 | [0, 4] | a[2] = 5 | prawa połowa |
| 2 | [3, 4] | a[3] = 7 | znaleziono |

Znaleziono na indeksie 3. Sprawdzono 2 elementy środkowe.

**Wyszukiwanie binarne i sortowanie przez scalanie.** Tryb „szukaj” wymaga danych niemalejących i pokazuje przedział, środkowy element i decyzję. Indeksy liczymy od zera. Tryb „sortuj” pokazuje kolejne scalenia dwóch uporządkowanych części; kolumny oznaczają wtedy lewą część, prawą część i wynik. Liczba faktycznych porównań nie jest tym samym co notacja O.

współczynnik zapełnienia tablicy haszującej

$$\alpha = \frac{n}{m}$$

n kluczy, m kubełków, m > 0; dla adresowania otwartego n ≤ m, dla łańcuchowania nie musi.

lemat o uściskach dłoni $\sum_{v \in V} \deg(v) = 2|E|$

Graf nieskierowany; pętla wnosi 2 do stopnia wierzchołka.

relaksacja krawędzi najkrótszej ścieżki

$$d(v) \leftarrow \min\{d(v), d(u) + w(u,v)\}$$

Element algorytmów najkrótszych ścieżek; Dijkstra wymaga wag nieujemnych, Bellman–Ford ma inne założenia.

## Symbole i jednostki

| symbol | znaczenie | jednostka |
|---|---|---|
| $a,\ b,\ c$ | stałe współczynniki modelu liczby operacji | — |
| $a$ | pierwszy argument NWD i NWW | 1 |
| $a_k$ | współczynnik k-tej potęgi wielomianu | 1 |
| $a_n$ | współczynnik najwyższej potęgi wielomianu | 1 |
| $b$ | drugi argument NWD i NWW | 1 |
| $b_0$ | końcowa wartość wielomianu | 1 |
| $b_k$ | akumulator Hornera po kroku k | 1 |
| $b_{k+1}$ | akumulator Hornera przed krokiem k | 1 |
| $b_n$ | początkowy akumulator Hornera | 1 |
| $c$ | stała | |

| symbol | znaczenie | jednostka |
|---|---|---|
| $d$ | aktualne oszacowanie odległości | |
| $E$ | zbiór krawędzi | |
| $f$ | badana funkcja kosztu | |
| $F_0$ | wyraz zerowy ciągu Fibonacciego | 1 |
| $F_1$ | pierwszy wyraz ciągu Fibonacciego | 1 |
| $F_n$ | n-ty wyraz ciągu Fibonacciego | 1 |
| $F_{n-1}$ | poprzedni wyraz ciągu Fibonacciego | 1 |
| $F_{n-2}$ | drugi poprzedni wyraz ciągu Fibonacciego | 1 |
| $g$ | funkcja porównawcza | |
| $k$ | maksymalna liczba iteracji | |

| symbol | znaczenie | jednostka |
|---|---|---|
| $m$ | liczba kubełków | 1 |
| $n$ | rozmiar danych wejściowych | — |
| $n$ | rozmiar danych | |
| $n$ | liczba kluczy | 1 |
| $n_0$ | próg rozmiaru | |
| $p$ | część obliczeń, którą można wykonać równolegle (od 0 do 1) | — |
| $s$ | przyspieszenie części równoległej | — |
| $S$ | przyspieszenie: czas sekwencyjny podzielony przez czas równoległy | — |
| $T$ | liczba operacji dla danych o rozmiarze n | — |
| $T$ | koszt algorytmu | |

| symbol | znaczenie | jednostka |
|---|---|---|
| $u$ | początek krawędzi | |
| $v$ | wierzchołek | |
| $V$ | zbiór wierzchołków | |
| $w$ | waga krawędzi | |
| $W$ | wielomian | 1 |
| $x$ | argument wielomianu | 1 |
| $\alpha$ | współczynnik zapełnienia | 1 |