Algorytmy i Struktury Danych, 1. ćwiczenia

2018-10-03

1 Plan zajęć

- przykłady algorytmów zachłannych, dziel i rządź, programowania dynamicznego,
- przykład z liczbami fibonacciego i wpływem określenia rozmiaru danych na złożoność,
- problemy 3.1–3.4 z Cormena

2 Algorytmy zachłanne, dziel i rządź, programowanie dynamiczne

- zachłanne: wydawanie monet, scalanie elementów (koszt scalenia x i y to $x + y$),
- dzieli i zwyciężaj: jednoczesne wyznaczenia min i max w ciągu używając minimalnej liczby porównań; mnożenie dużych liczb; inny problem: dany ciąg n różnych elementów ze zbioru $0..n$, należy wyznaczyć element którego nie ma w ciągu, jedyna operacja to $bit(i, k)$ zwracająca k-ty bit z i-tego elementu ciągu.
- dynamiczne: najdłuższy podciąg rosnący, fibonacci, scalanie sąsiadów, optymalne mnożenie macierzy,

Kontrprzykład dla algorytmu zachłannego dla scalania sąsiadów $(100, 99, 99, 100)$.

Mnożenie dużych liczb

Function Mult(a, b)

---

niech n oznacz długość liczb a, b
if $n \leq 1$ **then**
  użyj zwykłego mnożenia
else
  dzielimy (tekstowo) a i b na pary dwóch krótszych liczb (o $n/2$ cyfrach)
  niech $a = a_1\ a_2$ ($|a_1| = |a_2| = n/2$)
  niech $b = b_1\ b_2$ ($|b_1| = |b_2| = n/2$)
  $A = mult(a_1, b_1)$
  $B = mult(a_2, b_2)$
  $C = mult(a_1 + a_2, b_1 + b_2)$
  $D = C - (A + B)$ (co jest równoważne $D = a_1b_2 + a_2b_1$)
  return $A * 10^n + D * 10^{n/2} + B$;

Złożoność algorytmu:

$$T(n) = 3T\left(\frac{n}{2}\right) + O(n) = O(n^{\log_2 3}) \approx O(n^{1.58496})$$

3 Liczby Fibonacciego

Jak zmienia się złożność obliczeniowa w zależności od:

- dane wejściowe jako liczba binarna, dane wejściowe jak liczba unarna,
- algorytm dynamiczny i z mnożeniem macierzy.
- koszt operacji arytmetycznych stały, albo zależny od rozmiaru liczb.

Przydatna tożsamość:

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n = \begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix}$$

4 Cormen, zadania z 3. rozdziału

- 3.1 Asymptotyczne zachowanie wielomianów,
- 3.2 Względny rząd asymptotyczny,
- 3.3 Porządkowanie ze względu na rząd wielkości funkcji,
- 3.4 Własności notacji asymptycznej,

5 Temp

5.1 Definicje

$\Theta(g(n)) = \{f(n)$: istnieją dodatnie stałe c_1, c_2 i n_0, takie, że $0 \leq c_1g(n) \leq f(n) \leq c_2g(n)$ dla wszystkich $n \geq n_0\}$

$O(g(n)) = \{f(n)$: istnieją dodatnie stałe c i n_0, takie, że $f(n) \leq cg(n)$ dla wszystkich $n \geq n_0\}$
$\Omega(g(n)) = \{f(n)$: istnieją dodatnie stałe c i n_0, takie, że $0 \leq cg(n) \leq f(n)$ dla wszystkich $n \geq n_0\}$
$o(g(n)) = \{f(n)$: dla każdej dodatniej stałej $c > 0$ istnieje stała $n_0 > 0$, taka, że $0 \leq f(n) < cg(n)$ dla wszystkich $n \geq n_0\}$
$\omega(g(n)) = \{f(n)$: dla każdej dodatniej stałej $c > 0$ istnieje stała $n_0 > 0$, taka, że $0 \leq cg(n) < f(n)$ dla wszystkich $n \geq n_0\}$

5.2 Przydatne tożsamości

- $(a^m)^n = (a^n)^m$
- $a = b^{\log_b a}$
- $\log_b a^n = n \log_b a$
- $\log_b a = 1/\log_a b$
- $a^{\log_b c} = c^{\log_b a}$
- def: $\log^* n = \min\{i \geq 0 : \log^{(i)} n \leq 1\}$, przykłady: $\log^* 2 = 1$, $\log^* 4 = 2$, $\log^* 16 = 3$, $\log^* 65536 = 4$, $\log^*(2^{65536}) = 5$
- wzór Stirlinga:
  $$n! \leq \sqrt{2\pi n}(n/e)^n e^{\alpha_n}$$
  gdzie
  $$\frac{1}{12n+1} \leq \alpha_n \leq \frac{1}{12n}$$

5.3 Problem 3–1

[Cormen]
Niech

$$p(n) = \sum_{i=0}^{d} a_i n^i$$

gdzie $a_d > 0$, będzie wielomianem stopnia d zmiennej n i niech k będzie stałą. Korzystając z definicji notacji asymptotycznych, udowodnij następujące własności:

- jeśli $k \geq d$, to $p(n) = O(n^k)$
- jeśli $k \leq d$, to $p(n) = \Omega(n^k)$
- jeśli $k = d$, to $p(n) = \Theta(n^k)$
- jeśli $k > d$, to $p(n) = o(n^k)$
- jeśli $k < d$, to $p(n) = \omega(n^k)$

5.4 Problem 3–2

[Cormen]
Porównaj względem notacji O, o, Ω, ω, Θ funkcje:

$\log^k n$	$<$	n^ϵ
n^k	$<$	c^n
$\sqrt{n}$	nie da się porównać	$n^{\sin n}$
2^n	$>$	$2^{n/2}$
$n^{\log c}$	$=$	$c^{\log n}$
$\log n!$	$=$	$\log n^n = O(n \log n)$

Przymij, że $k \geq 1$, $\epsilon > 0$ i $c > 1$ są stałymi.

5.5 Problem 3–3

[Cormen]
Uporządkuj następujące funkcje ze względu na ich rząd wielkości:

- 1
- $n^{1/\log n} = n^{\log_n 2} = 2$
- $\log \log^* n$
- $\log^*(\log n) = O(\log^* n)$
- $\log^* n$
- $2^{\log^* n}$
- $\log \log n$
- $\sqrt{\log n}$
- $\log n$
- $\log^2 n$
- $2^{\sqrt{2 \log n}}$
- $(\sqrt{2})^{\log n} = \sqrt{n}$
- $2^{\log n} = n$
- n
- $\log(n!)$ $(\Omega(n),$ $O(\log n^n) = O(n \log n))$
- $n \log n$
- n^2
- $4^{\log n} = n^2$
- n^3
- $(\log n)!$ $(\Omega(n),$ $O((\log n)^{\log n}) = O(n^{\log \log n})$
- $n^{\log \log n}$
- $(\log n)^{\log n} = n^{\log \log n}$
- $(3/2)^n$
- 2^n
- $n \cdot 2^n$
- e^n
- $n!$ $(\Omega(2^n)$, $O(n^n))$
- $(n+1)! = (n+1) \cdot n!$
- 2^{2^n}
- $2^{2^{n+1}} = (2^{2^n})^2$

5.6 Problem 3–4

[Cormen] Niech $f(n)$ i $g(n)$ będą funkcjami asymptotycznymi dodatnimi. Udowodnij lub obal każde z tych następujących stwierdzeń:

- $f(n) = O(g(n))$ implikuje $g(n) = O(f(n))$ FAŁSZ,
- $f(n) + g(n) = \Theta(min(f(n), g(n)))$ FAŁSZ,

- $f(n) = O(g(n))$ implikuje $\log(f(n)) = O(\log(g(n)))$, gdzie $\log(g(n)) \geq 1$ i $f(n) \geq 1$ dla wszystkich dostatecznie dużych n PRAWDA,
- $f(n) = O(g(n))$ implikuje $2^{f(n)} = O(2^{g(n)})$ FAŁSZ ($f(n) = 2n$, $g(n) = n$),
- $f(n) = O((f(n))^2)$ FAŁSZ (np. $f(n) = 1/n$),
- $f(n) = O(g(n))$ implikuje $g(n) = \Omega(f(n))$ PRAWDA,
- $f(n) = \Theta(f(n/2))$ FAŁSZ (np. $f(n) = 2^n$),
- $f(n) + o(f(n)) = \Theta(f(n))$ PRAWDA.

5.7 Twierdzenie 4–1

(Twierdzenie o rekurencji uniwersalnej)

Niech $a \geq 1$ i $b > 1$ będą stałymi, niech $f(n)$ będzie pewną funkcją i niech $T(n)$ będzie zdefiniowane dla nieujemnych liczb całkowitych przez rekurencję:

$$T(n) = aT(n/b) + f(n)$$

gdzie n/b oznacza $\lfloor n/b \rfloor$ lub $\lceil n/b \rceil$. Wtedy funkcja $T(n)$ może być ograniczona asymptotycznie w następujący sposób:

- jeśli $f(n) = O(n^{\log_b a - \epsilon})$ dla pewnej stałej $\epsilon > 0$, to $T(n) = \Theta(n^{\log_b a})$.
- jeśli $f(n) = \Theta(n^{\log_b a})$, to $T(n) = \Theta(n^{\log_b a} \log n)$.
- jeśli $f(n) = \Omega(n^{\log_b a + \epsilon})$ dla pewnej stałej $\epsilon > 0$ i jeśli $af(n/b) \leq cf(n)$ dla pewnej stałej $c < 1$ i wszystkich dostatecznie dużych n, to $T(n) = \Theta(f(n))$.