Literatura

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. *Wprowadzenie do algorytmów.*

Literatura

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. *Wprowadzenie do algorytmów.*
- A.V. Aho, J.E. Hopcroft, J.D. Ullman. *Projektowanie i analiza algorytmów komputerowych.*

Literatura

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. *Wprowadzenie do algorytmów.*
- A.V. Aho, J.E. Hopcroft, J.D. Ullman. *Projektowanie i analiza algorytmów komputerowych.*
- Knuth Donald E. *Sztuka programowania.*
- Niklaus Wirth. *Algorytmy + struktury danych = programy.*
- L. Banachowski, K. Diks, W. Rytter. *Algorytmy i struktury danych*
- . . .

Insertion-Sort

```
Insertion-Sort(A)
  for j ← 2 to length[A] do
```

Insertion-Sort

`Insertion-Sort(`A`)`	
`for` $j \leftarrow 2$ `to` $length[A]$ `do`	n
$\quad key \leftarrow A[j]$	$n-1$
$\quad \triangleright$ `Wstaw` $A[j]$ `do posortowanego`	
$\quad \triangleright$ `ciągu` $A[1] \ldots A[j-1]$	
$\quad i \leftarrow j-1$	$n-1$
$\quad$ `while` $i > 0$ `and` $A[i] > key$ `do`	$\sum_{j=2}^{n} t_j$
$\qquad A[i+1] \leftarrow A[i]$	$\sum_{j=2}^{n}(t_j - 1)$
$\qquad i \leftarrow i-1$	$\sum_{j=2}^{n}(t_j - 1)$
$\quad A[i+1] \leftarrow key$	$n-1$

Insertion-Sort

Insertion-Sort(A)	
for $j \leftarrow 2$ to $length[A]$ do	n
$key \leftarrow A[j]$	$n-1$
$\triangleright$ Wstaw $A[j]$ do posortowanego	
$\triangleright$ ciągu $A[1] \ldots A[j-1]$	
$i \leftarrow j-1$	$n-1$
while $i > 0$ and $A[i] > key$ do	$\sum_{j=2}^{n} t_j$
$A[i+1] \leftarrow A[i]$	$\sum_{j=2}^{n}(t_j-1)$
$i \leftarrow i-1$	$\sum_{j=2}^{n}(t_j-1)$
$A[i+1] \leftarrow key$	$n-1$

$$T(n) = c_1 n + (c_2 + c_4 + c_8)(n-1) + c_5 \sum_{j=2}^{n} t_j + (c_6 + c_7) \sum_{j=2}^{n} (t_j - 1)$$

Insertion-Sort

$$T(n) = c_1 n + (c_2 + c_4 + c_8)(n-1) + c_5 \sum_{j=2}^{n} t_j + (c_6 + c_7) \sum_{j=2}^{n} (t_j - 1)$$

Dla posortowanego *rosnąco* A mamy $t_j = 1$.

Insertion-Sort

$$T(n) = c_1 n + (c_2 + c_4 + c_8)(n-1) + c_5 \sum_{j=2}^{n} t_j + (c_6 + c_7) \sum_{j=2}^{n} (t_j - 1)$$

Dla posortowanego *rosnąco* A mamy $t_j = 1$.

$$\begin{aligned} T(n) &= c_1 n + (c_2 + c_4)(n-1) + c_5(n-1) + c_8(n-1) \\ &= (c_1 + c_2 + c_3 + c_4 + c_5 + c_8)n - (c_2 + c_4 + c_5 + c_8) \end{aligned}$$

Insertion-Sort

$$T(n) = c_1 n + (c_2 + c_4 + c_8)(n-1) + c_5 \sum_{j=2}^{n} t_j + (c_6 + c_7) \sum_{j=2}^{n} (t_j - 1)$$

Dla posortowanego *malejąco* A mamy $t_j = j$.

Insertion-Sort

$$T(n) = c_1 n + (c_2 + c_4 + c_8)(n-1) + c_5 \sum_{j=2}^{n} t_j + (c_6 + c_7) \sum_{j=2}^{n} (t_j - 1)$$

Dla posortowanego *malejąco* A mamy $t_j = j$.

$$\textstyle\sum_{j=2}^{n} j = \frac{n(n+1)}{2} - 1, \quad \sum_{j=2}^{n} (j-1) = \frac{n(n-1)}{2}$$

$$\begin{aligned} T(n) = \quad & c_1 n + (c_2 + c_4 + c_8)(n-1) + c_5 \left(\frac{n(n+1)}{2} - 1 \right) \\ & + (c_6 + c_7) \left(\frac{n(n-1)}{2} \right) \\ = \quad & \frac{c_5 + c_6 + c_7}{2} n^2 + \left(c_1 + c_2 + c_4 + c_8 + \frac{c_5}{2} - \frac{c_6 + c_7}{2} \right) n \\ & - (c_2 + c_4 + c_8 + c_5) \end{aligned}$$

Insertion-Sort

$$T(n) = c_1 n + (c_2 + c_4 + c_8)(n-1) + c_5 \sum_{j=2}^{n} t_j + (c_6 + c_7) \sum_{j=2}^{n} (t_j - 1)$$

Dla posortowanego *malejąco* A mamy $t_j = j$.

$$\textstyle\sum_{j=2}^{n} j = \frac{n(n+1)}{2} - 1, \quad \sum_{j=2}^{n} (j-1) = \frac{n(n-1)}{2}$$

$$\begin{aligned} T(n) = \;& c_1 n + (c_2 + c_4 + c_8)(n-1) + c_5 \left(\tfrac{n(n+1)}{2} - 1 \right) \\ & + (c_6 + c_7) \left(\tfrac{n(n-1)}{2} \right) \\ = \;& \tfrac{c_5 + c_6 + c_7}{2} n^2 + \left(c_1 + c_2 + c_4 + c_8 + \tfrac{c_5}{2} - \tfrac{c_6 + c_7}{2} \right) n \\ & - (c_2 + c_4 + c_8 + c_5) \end{aligned}$$

Jest to przypadek najgorszy, bo zawsze $t_j \leq j$.

Rodzaje złożoności

- *pesymistyczna:* najgorszy przypadek dla każdego rozmiaru danych

Rodzaje złożoności

- *pesymistyczna:* najgorszy przypadek dla każdego rozmiaru danych
- *oczekiwana:* średnia wartość złożoności (z uwzględnieniem prawdopodobieństwa poszczególnych przypadków)

Rodzaje złożoności

- *pesymistyczna:* najgorszy przypadek dla każdego rozmiaru danych
- *oczekiwana:* średnia wartość złożoności (z uwzględnieniem prawdopodobieństwa poszczególnych przypadków)

Ze względu na rodzaj zasobu:

- *czasowa*

Rodzaje złożoności

- *pesymistyczna:* najgorszy przypadek dla każdego rozmiaru danych
- *oczekiwana:* średnia wartość złożoności (z uwzględnieniem prawdopodobieństwa poszczególnych przypadków)

Ze względu na rodzaj zasobu:

- *czasowa*
- *pamięciowa*

Notacja Asymptotyczna

Dla danej funkcji $g(n)$:

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

Notacja Asymptotyczna

Dla danej funkcji $g(n)$:

$$\Theta(g(n)) = \{f(n) : \text{istnieją dodatnie stałe } c_1,\ c_2 \text{ i } n_0 \text{ takie, że } 0 \le c_1 g(n) \le f(n) \le c_2 g(n) \text{ dla wszystkich } n \ge n_0\}$$

Notacja: „$f(n) = \Theta(g(n))$” oznacza „$f(n) \in \Theta(g(n))$”

Przykład

$$\frac{1}{2}n^2 - 3n = \Theta(n^2)$$

Przykład

$$\frac{1}{2}n^2 - 3n = \Theta(n^2)$$

$$c_1 n^2 \leq \frac{1}{2}n^2 - 3n \leq c_2 n^2$$

Przykład

$$\frac{1}{2}n^2 - 3n = \Theta(n^2)$$

$$c_1 n^2 \leq \frac{1}{2}n^2 - 3n \leq c_2 n^2$$

$$c_1 \leq \frac{1}{2} - \frac{3}{n} \leq c_2$$

Przykład

$$\frac{1}{2}n^2 - 3n = \Theta(n^2)$$

$$c_1 n^2 \leq \frac{1}{2}n^2 - 3n \leq c_2 n^2$$

$$c_1 \leq \frac{1}{2} - \frac{3}{n} \leq c_2$$

Wybieramy $c_1 = 1/14$, $c_2 = 1/2$, $n_0 = 7$.

Notacja $O()$, $\Omega()$

Dla danej funkcji $g(n)$:

$$O(g(n)) = \{f(n) : \text{istnieją dodatnie stałe } c \text{ i } n_0 \text{ takie, że } 0 \leq f(n) \leq cg(n) \text{ dla wszystkich } n \geq n_0\}$$

Notacja $O()$, $\Omega()$

Dla danej funkcji $g(n)$:

$$O(g(n)) = \{f(n) : \text{istnieją dodatnie stałe } c \text{ i } n_0 \text{ takie, że } 0 \le f(n) \le cg(n) \text{ dla wszystkich } n \ge n_0\}$$

$$\Omega(g(n)) = \{f(n) : \text{istnieją dodatnie stałe } c \text{ i } n_0 \text{ takie, że } 0 \le cg(n) \le f(n) \text{ dla wszystkich } n \ge n_0\}$$

Przykłady

- $8n^2 + 3n = O(n^2)$
- $1000n = O(n^3)$
- $\frac{1}{1000}n^3 = \Omega(n^3)$
- $\frac{1}{1000}n^3 = \Omega(1000n^2)$

Twierdzenie

Twierdzenie. Dla każdych dwóch funkcji $f(n)$ i $g(n)$ zachodzi: $f(n) = \Theta(g(n))$ wtedy i tylko wtedy gdy $f(n) = O(g(n))$ oraz $f(n) = \Omega(g(n))$.

Notacja $o(), \omega()$

Dla danej funkcji $g(n)$:

$$o(g(n)) = \{f(n) : \text{dla każdej stałej } c > 0 \text{ istnieje stała } n_0 > 0 \text{ taka, że } 0 \le f(n) < cg(n) \text{ dla wszystkich } n \ge n_0\}$$

Notacja $o(), \omega()$

Dla danej funkcji $g(n)$:

$o(g(n))$ = $\{f(n)$: dla każdej 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 stałej $c > 0$ istnieje stała $n_0 > 0$
taka, że $0 \leq cg(n) < f(n)$
dla wszystkich $n \geq n_0\}$

Przykłady

- $8n^2 + 3n \neq o(n^2)$
- $1000n = o(n^2)$
- $n^3 \neq \omega(n^3)$
- $\frac{1}{1000}n^3 = \omega(1000n^2)$

Własności

Przechodniość:

- $f(n) = \Theta(g(n))$ i $g(n) = \Theta(h(n))$ implikuje $f(n) = \Theta(h(n))$

Własności

Przechodniość:

- $f(n) = \Theta(g(n))$ i $g(n) = \Theta(h(n))$ implikuje $f(n) = \Theta(h(n))$
- $f(n) = O(g(n))$ i $g(n) = O(h(n))$ implikuje $f(n) = O(h(n))$
- $f(n) = \Omega(g(n))$ i $g(n) = \Omega(h(n))$ implikuje $f(n) = \Omega(h(n))$
- $f(n) = o(g(n))$ i $g(n) = o(h(n))$ implikuje $f(n) = o(h(n))$
- $f(n) = \omega(g(n))$ i $g(n) = \omega(h(n))$ implikuje $f(n) = \omega(h(n))$

Własności

Przechodniość:

- $f(n) = \Theta(g(n))$ i $g(n) = \Theta(h(n))$ implikuje $f(n) = \Theta(h(n))$
- $f(n) = O(g(n))$ i $g(n) = O(h(n))$ implikuje $f(n) = O(h(n))$
- $f(n) = \Omega(g(n))$ i $g(n) = \Omega(h(n))$ implikuje $f(n) = \Omega(h(n))$
- $f(n) = o(g(n))$ i $g(n) = o(h(n))$ implikuje $f(n) = o(h(n))$
- $f(n) = \omega(g(n))$ i $g(n) = \omega(h(n))$ implikuje $f(n) = \omega(h(n))$

Zwrotność:

- $f(n) = \Theta(f(n))$
- $f(n) = O(f(n))$
- $f(n) = \Omega(f(n))$

Własności

Symetria:

- $f(n) = \Theta(g(n))$ wtedy i tylko wtedy, gdy $g(n) = \Theta(f(n))$

Własności

Symetria:

- $f(n) = \Theta(g(n))$ wtedy i tylko wtedy, gdy $g(n) = \Theta(f(n))$

Symetria transpozycyjna:

- $f(n) = O(g(n))$ wtedy i tylko wtedy, gdy $g(n) = \Omega(f(n))$
- $f(n) = o(g(n))$ wtedy i tylko wtedy, gdy $g(n) = \omega(f(n))$

Dziel i zwyciężaj

Dziel: Dzielimy problem na podproblemy.

Dziel i zwyciężaj

Dziel: Dzielimy problem na podproblemy.

Zwyciężaj: Rozwiązujemy podproblemy rekurencyjnie lub (jeśli są małego rozmiaru) metodami bezpośrednimi.

Dziel i zwyciężaj

Dziel: Dzielimy problem na podproblemy.

Zwyciężaj: Rozwiązujemy podproblemy rekurencyjnie lub (jeśli są małego rozmiaru) metodami bezpośrednimi.

Połącz: Łączymy rozwiązania podproblemów, aby otrzymać rozwiązanie całego problemu.

Dziel i zwyciężaj

Dziel: Dzielimy problem na podproblemy.

Zwyciężaj: Rozwiązujemy podproblemy rekurencyjnie lub (jeśli są małego rozmiaru) metodami bezpośrednimi.

Połącz: Łączymy rozwiązania podproblemów, aby otrzymać rozwiązanie całego problemu.

Sortowanie przez scalanie

Dziel: Dzielimy n-elementowy ciąg na dwa podciągi po $n/2$ elementów.

Zwyciężaj: Sortujemy każdy z obu podciągów rekurencyjnie.

Połącz: Łączymy posortowane podciągi w jeden posortowany ciąg.

Merge-Sort

```
Merge-Sort(A,p,r)
if p < r then
    q ← ⌊(p + r)/2⌋
    Merge-Sort(A, p, q)
    Merge-Sort(A, q + 1, r)
    Merge(A, p, q, r)
```

Merge-Sort

```
Merge-Sort(A,p,r)
if p < r then
    q ← ⌊(p + r)/2⌋
    Merge-Sort(A, p, q)
    Merge-Sort(A, q + 1, r)
    Merge(A, p, q, r)
```

Dla tablicy $A = \langle A[1], A[2], \ldots A[n]\rangle$ wywołujemy:
`Merge-Sort(`$A, 1, length[A]$`)`,
gdzie $length[A] = n$.

Merge

```
Merge(A,p,q,r)
i ← p
j ← q + 1
k ← 1
while i ≤ q and j ≤ r do
    if A[i] ≤ A[j] then
        A'[k] = A[i]
        i ← i + 1
    else
        A'[k] = A[j]
        j ← j + 1
    k ← k + 1
...
```

Merge

```
...
while i ≤ q do
    A'[k] = A[i]
    i ← i + 1
    k ← k + 1
while j ≤ r do
    A'[k] = A[j]
    j ← j + 1
    k ← k + 1
for k ← 1 to r − p + 1 do
    A[p + k − 1] ← A'[k]
```

Czas: $\Theta(r - p + 1)$

Analiza Merge-Sort

Dziel: Znajdujemy środek przedziału. Czas: $D(n) = O(1)$

Zwyciężaj: Rozwiązujemy dwa podproblemy rozmiaru $n/2$.
Czas: $2T(n/2)$.

Połącz: `Merge.` Czas: $C(n) = \Theta(n)$

$$T(n) = \begin{cases} \Theta(1) & \text{dla } n = 1 \\ T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + \Theta(n) & \text{dla } n > 1 \end{cases}$$

Analiza Merge-Sort

Dziel: Znajdujemy środek przedziału. Czas: $D(n) = O(1)$

Zwyciężaj: Rozwiązujemy dwa podproblemy rozmiaru $n/2$.
Czas: $2T(n/2)$.

Połącz: `Merge.` Czas: $C(n) = \Theta(n)$

$$T(n) = \begin{cases} \Theta(1) & \text{dla } n = 1 \\ T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + \Theta(n) & \text{dla } n > 1 \end{cases}$$

Pomijanie szczegółów:

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

Rekurencje

Metoda podstawiania. Przykład.

$$T(n) = 2T(\lfloor n/2 \rfloor) + n$$

Rekurencje

Metoda podstawiania. Przykład.

$$T(n) = 2T(\lfloor n/2 \rfloor) + n$$

Zgadujemy, że $T(n) = O(n \lg n)$.
(Uwaga: „$\lg x$” oznacza „$\log_2 x$”.)
Musimy udowodnić, że $T(n) \leq cn \log n$, dla pewnej stałej $c > 0$.

$$\begin{aligned} T(n) &\leq 2(c\lfloor n/2 \rfloor \lg \lfloor n/2 \rfloor) + n \quad \text{(zał.ind.)} \\ &\leq cn \lg(n/2) + n \\ &= cn \lg n - cn \lg 2 + n \\ &= cn \lg n - cn + n \\ &\leq cn \lg n \quad \text{(jeśli } c \geq 1\text{)} \end{aligned}$$

Rekurencje

Problem z warunkiem brzegowym $T(1) = 1$:

$$c1 \lg 1 = 0$$

Wyznaczamy:

$$T(2) = 2T(1) + 2 = 4$$

oraz

$$T(3) = 2T(1) + 3 = 5$$

jako nowe warunki brzegowe.
Wybieramy c tak aby $T(2) \leq c2 \lg 2$ i $T(3) \leq c3 \lg 3$.
(Wystarczy dowolne $c \geq 2$.)

Rekurencje

$$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1$$

Rekurencje

$$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1$$

Zgadujemy, że $T(n) = O(n)$.
Musimy pokazać, że $T(n) \leq cn$ dla pewnej stałej c.

$$\begin{aligned} T(n) &\leq c\lfloor n/2 \rfloor + c\lceil n/2 \rceil + 1 \\ &= cn + 1 \quad \Leftarrow \text{problem!} \end{aligned}$$

Rekurencje

Rozwiązanie: Zakładamy, że $T(n) \leq cn - b$, gdzie $b \geq 0$ jest stałą.

$$\begin{aligned} T(n) &\leq (c\lfloor n/2 \rfloor - b) + (c\lceil n/2 \rceil - b) + 1 \\ &= cn - 2b + 1 \\ &\leq cn - b \quad \text{(gdy } b \geq 1\text{)} \end{aligned}$$

Rekurencje - zamiana zmiennych

$$T(n) = 2T(\sqrt{n}) + \lg n$$

Rekurencje - zamiana zmiennych

$$T(n) = 2T(\sqrt{n}) + \lg n$$

Podstawiamy: $m = \lg n$.

$$T(2^m) = 2T(2^{m/2}) + m$$

Podstawiamy: $S(m) = T(2^m)$.

$$S(m) = 2S(m/2) + m$$
$$S(m) = O(m \lg m) \quad \text{(To już wiemy!)}$$
$$T(n) = T(2^m) = S(m) = O(m \lg m) = O(\lg n \lg \lg n)$$

Rekurencje – metoda iteracyjna

$$T(n) \quad = \quad 3T(n/4) + n$$

Rekurencje – metoda iteracyjna

$$T(n) \quad = \quad 3T(n/4) + n$$

$$\begin{aligned} T(n) &= n + 3T(n/4) \\ &= n + 3(n/4 + 3T(n/16)) \\ &= n + 3(n/4 + 3(n/16 + 3T(n/64))) \\ &= n + 3n/4 + 9n/16 + 27T(n/64) \end{aligned}$$

Rekurencje – metoda iteracyjna

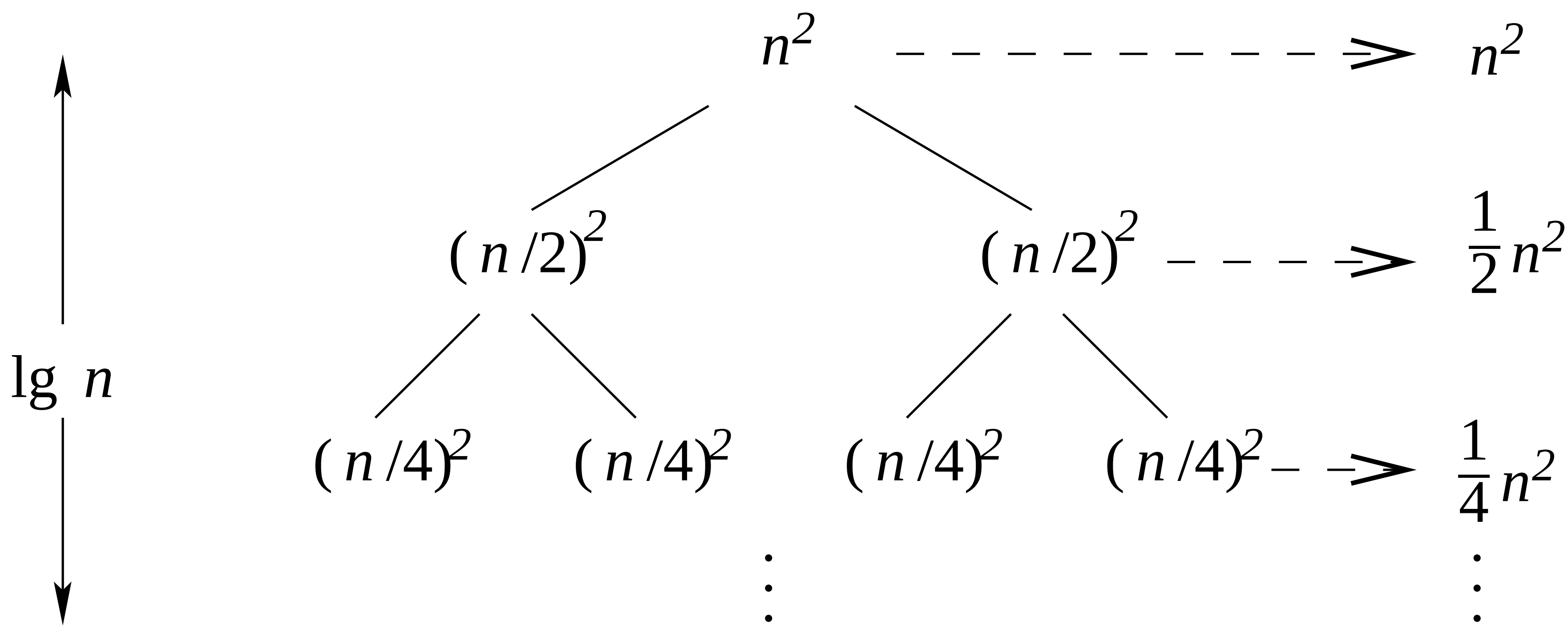
$$T(n) = 3T(n/4) + n$$

$$\begin{aligned} T(n) &= n + 3T(n/4) \\ &= n + 3(n/4 + 3T(n/16)) \\ &= n + 3(n/4 + 3(n/16 + 3T(n/64))) \\ &= n + 3n/4 + 9n/16 + 27T(n/64) \end{aligned}$$

$$\begin{aligned} T(n) &\le n + 3n/4 + 9n/16 + 27n/64 + \ldots + 3^{\log_4 n}\Theta(1) \\ &\le n\sum_{i=0}^{\infty}(3/4)^i + \Theta(n^{\log_4 3}) \\ &\le 4n + o(n) \\ &= O(n) \end{aligned}$$

drzewo rekursji

$$T(n) = 2T(n/2) + n^2$$



W sumie: $\Theta(n^2)$

Twierdzenie o rekurencji uniwersalnej

Tw. Niech $a \geq 1$,$b > 1$ – stałe, $f(n)$ – funkcja, oraz:

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

gdzie n/b oznacza $\lceil n/b \rceil$ lub $\lfloor n/b \rfloor$. Wtedy:

Twierdzenie o rekurencji uniwersalnej

Tw. Niech $a \geq 1$, $b > 1$ – stałe, $f(n)$ – funkcja, oraz:

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

gdzie n/b oznacza $\lceil n/b \rceil$ lub $\lfloor n/b \rfloor$. Wtedy:

1. Jeśli $f(n) = O(n^{\log_b a - \epsilon})$ dla stałej $\epsilon > 0$, to $T(n) = \Theta(n^{\log_b a})$.

Twierdzenie o rekurencji uniwersalnej

Tw. Niech $a \geq 1$,$b > 1$ – stałe, $f(n)$ – funkcja, oraz:

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

gdzie n/b oznacza $\lceil n/b \rceil$ lub $\lfloor n/b \rfloor$. Wtedy:

1. Jeśli $f(n) = O(n^{\log_b a - \epsilon})$ dla stałej $\epsilon > 0$, to $T(n) = \Theta(n^{\log_b a})$.
2. Jeśli $f(n) = \Theta(n^{\log_b a})$, to $T(n) = \Theta(n^{\log_b a} \lg n)$.

Twierdzenie o rekurencji uniwersalnej

Tw. Niech $a \geq 1$,$b > 1$ – stałe, $f(n)$ – funkcja, oraz:

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

gdzie n/b oznacza $\lceil n/b \rceil$ lub $\lfloor n/b \rfloor$. Wtedy:

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

Szkic dowodu

Zakładamy, że $n = b^k$, dla k – naturalnego.

$$
\begin{aligned}
T(n) &= f(n) + aT(n/b) \\
&= f(n) + a(f(n/b) + aT(n/b^2)) \\
&= a^0 f(n/b^0) + a^1 f(n/b^1) + a^2 T(n/b^2) \\
&= a^0 f(n/b^0) + \ldots + a^{\log_b n - 1} f(n/b^{\log_b n - 1}) + a^{\log_b n} T(1) \\
&= \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j) + \Theta(n^{\log_b a})
\end{aligned}
$$

Niech $g(n) = \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j)$.

szkic dowodu

Jeśli $f(n) = O(n^{\log_b a - \epsilon})$, to $f(n/b^j) = O((n/b^j)^{\log_b a - \epsilon})$ czyli:
$g(n) = O\left(\sum_{j=0}^{\log_b n - 1} a^j (n/b^j)^{\log_b a - \epsilon}\right)$

$$\begin{aligned}
\sum_{j=0}^{\log_b n - 1} a^j (n/b^j)^{\log_b a - \epsilon} &= n^{\log_b a - \epsilon} \sum_{j=0}^{\log_b n - 1} \left(\frac{ab^{\epsilon}}{b^{\log_b a}}\right)^j \\
&= n^{\log_b a - \epsilon} \sum_{j=0}^{\log_b n - 1} (b^{\epsilon})^j \\
&= n^{\log_b a - \epsilon} \left(\frac{b^{\epsilon \log_b n} - 1}{b^{\epsilon} - 1}\right) \\
&= n^{\log_b a - \epsilon} \left(\frac{n^{\epsilon} - 1}{b^{\epsilon} - 1}\right) \\
&= n^{\log_b a - \epsilon} O(n^{\epsilon}) \\
&= O(n^{\log_b a})
\end{aligned}$$

Przypadek 1: $T(n) = g(n) + \Theta(n^{\log_b a}) = \Theta(n^{\log_b a})$.

szkic dowodu

Jeśli $f(n) = \Theta(n^{\log_b a})$, to $f(n/b^j) = \Theta((n/b^j)^{\log_b a})$ czyli:

$$g(n) = \Theta\left(\sum_{j=0}^{\log_b n-1} a^j (n/b^j)^{\log_b a}\right)$$

$$\begin{aligned}
\textstyle\sum_{j=0}^{\log_b n-1} a^j (n/b^j)^{\log_b a} &= n^{\log_b a} \textstyle\sum_{j=0}^{\log_b n-1} \left(\frac{a}{b^{\log_b a}}\right)^j \\
&= n^{\log_b a} \textstyle\sum_{j=0}^{\log_b n-1} 1 \\
&= n^{\log_b a} \log_b n
\end{aligned}$$

Przypadek 2: $T(n) = g(n) + \Theta(n^{\log_b a}) = \Theta(n^{\log_b a} \log_b n)$.

szkic dowodu

Załóżmy, że $af(n/b) \le cf(n)$ dla stałej c, $0 < c < 1$, i wszystkich $n > b$. Wtedy

$$\begin{aligned} g(n) &\le \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j) \\ &\le \sum_{j=0}^{\log_b n - 1} c^j f(n) \\ &\le f(n) \sum_{j=0}^{\infty} c^j \\ &= f(n) \left(\frac{1}{1-c}\right) \end{aligned}$$

Z drugiej strony

$g(n) \ge f(n) + ($dodatnia reszta sumy$) = \Omega(f(n))$.

Ponadto: $af(n/b) \le cf(n)$ implikuje $f(n) = \Omega(n^{\log_b a + \epsilon})$. (ćw.)

Przypadek 3: $T(n) = g(n) + \Theta(n^{\log_b a}) = \Theta(f(n))$

Korzystanie z tw. o rekursji uniwersalnej

1. $T(n) = 9T(n/3) + n$
   $a = 9$, $b = 3$, $f(n) = n$, $n^{\log_b a} = n^2$,
   $f(n) = O(n^{\log_b a - \epsilon})$ dla $\epsilon = 1 > 0$.
   Przypadek 1: $T(n) = \Theta(n^{\log_b a}) = \Theta(n^2)$.

Korzystanie z tw. o rekursji uniwersalnej

1. $T(n) = 9T(n/3) + n$
   $a = 9$, $b = 3$, $f(n) = n$, $n^{\log_b a} = n^2$,
   $f(n) = O(n^{\log_b a - \epsilon})$ dla $\epsilon = 1 > 0$.
   Przypadek 1: $T(n) = \Theta(n^{\log_b a}) = \Theta(n^2)$.
2. $T(n) = T(2n/3) + 1$
   $a = 1$, $b = 3/2$, $f(n) = 1$, $n^{\log_b a} = n^{\log_{3/2} 1} = n^0 = 1$,
   $f(n) = \Theta(1) = \Theta(n^{\log_b a})$.
   Przypadek 2: $T(n) = \Theta(\lg n)$.

Korzystanie z tw. o rekursji uniwersalnej

1. $T(n) = 9T(n/3) + n$
   $a = 9$, $b = 3$, $f(n) = n$, $n^{\log_b a} = n^2$,
   $f(n) = O(n^{\log_b a - \epsilon})$ dla $\epsilon = 1 > 0$.
   Przypadek 1: $T(n) = \Theta(n^{\log_b a}) = \Theta(n^2)$.
2. $T(n) = T(2n/3) + 1$
   $a = 1$, $b = 3/2$, $f(n) = 1$, $n^{\log_b a} = n^{\log_{3/2} 1} = n^0 = 1$,
   $f(n) = \Theta(1) = \Theta(n^{\log_b a})$.
   Przypadek 2: $T(n) = \Theta(\lg n)$.
3. $T(n) = 3T(n/4) + n \lg n$
   $a = 3$, $b = 4$, $f(n) = n \lg n$, $n^{\log_b a} = n^{\log_4 3} = O(n^{0,793})$,
   $f(n) = \Omega(n^{\log_4 3 + \epsilon})$ dla $\epsilon \approx 0,2$. Dla dużych n:
   $af(n/b) = 3(n/4)\lg(n/4) \leq (3/4) n \lg n = cf(n)$ dla
   $c = 3/4$.
   Przypadek 3: $T(n) = \Theta(n \lg n)$.

korzystanie z tw. o rekursji uniwersalnej

$T(n) = 2T(n/2) + n \lg n$

$a = 2$, $b = 2$, $f(n) = n \lg n$, $n^{\log_b a} = n$.

Ale dla każdego $\epsilon > 0$: $n \lg n \neq \Omega(n^{\log_b a + \epsilon}) = \Omega(n^{1+\epsilon})$

Nie zachodzi przypadek 3 (ani żaden inny).