Algorytmy i struktury danych

Lista 3

Zad 1.

Rekurencja uniwersalna

$T(n) = aT(\frac{n}{b}) + f(n)$, gdzie $a \geq 1$, $b > 1$, $f(n)$ - asymptotycznie dodatnia

A) Jeśli $f(n) = O(n^{\log_b a - \varepsilon})$, dla pewnego $\varepsilon > 0$
to $T(n) = \Theta(n^{\log_b a})$

B) Jeśli $f(n) = \Theta(n^{\log_b a})$
to $T(n) = \Theta(n^{\log_b a} \log n)$

C) Jeśli $f(n) = \Omega(n^{\log_b a + \varepsilon})$, dla pewnego $\varepsilon > 0$
oraz $af(\frac{n}{b}) \leq c f(n)$, dla pewnego $c < 1$ i dostatecznie dużych n
to $T(n) = \Theta(f(n))$

---

a) $T(n) = 5T(\frac{n}{3}) + n$

$a = 5 \quad b = 3 \quad f(n) = n$

$\log_3 5 \approx 1{,}46 \Rightarrow \varepsilon \approx 0{,}46$

$f(n) = n = O(n^{\log_3 5 - 0,\ \ })$ postać $O(n^{\log_b a - \varepsilon})$

$\Downarrow$

$T(n) = \Theta(n^{\log_3 5}) \approx \Theta(n^{1,46})$

b) $T(n) = 4T(\frac{n}{2}) + n^2$

$a = 4 \quad b = 2 \quad f(n) = n^2$

$\log_b a = \log_2 4 = 2$

$f(n) = n^2 = \Theta(n^{\log_2 4}) = \Theta(n^{\log_b a})$

$\Downarrow$

$T(n) = \Theta(n^{\log_2 4} \log n) = \Theta(n^2 \log n)$

c) $T(n) = 9T\left(\frac{n}{3}\right) + n^2$

$a = 9 \quad b = 3 \quad f(n) = n^2$

$\log_b a = \log_3 9 = 2$

$f(n) = n^2 = \Theta(n^{\log_3 9}) = \Theta(n^{\log_b a})$

$\Downarrow$

$T(n) = \Theta(n^{\log_3 9} \log n) = \Theta(n^2 \log n)$

d) $T(n) = 6T\left(\frac{n}{3}\right) + n^2$

$a = 6 \quad b = 3 \quad f(n) = n^2$

$\log_b a = \log_3 6 \approx 1{,}63 \Rightarrow \varepsilon \approx 0{,}37$

$f(n) = n^2 = \Omega(n^{\log_3 6 + 0{,}37})$

$a f\left(\frac{n}{b}\right) \leq c\, f(n) \quad c < 1$

$6 \cdot \left(\frac{n}{3}\right)^2 = 6 \frac{n^2}{9} \leq c \cdot n^2 \Rightarrow c = \frac{6}{9} = \frac{2}{3} < 1$

$\Downarrow$

$T(n) = \Theta(f(n)) = \Theta(n^2)$

e) $T(n) = 3T\left(\frac{n}{3}\right) + n$

$a = 3 \quad b = 3 \quad f(n) = n$

$\log_b a = \log_3 3 = 1$

$f(n) = n = \Theta(n^{\log_3 3}) = \Theta(n)$

$\Downarrow$

$T(n) = \Theta(n^{\log_3 3} \log n) = \Theta(n \log n)$

f) $T(n) = 5T\left(\frac{n}{2}\right) + n^2$

$a = 5 \quad b = 2 \quad f(n) = n^2$

$\log_b a = \log_2 5 \approx 2{,}32 \Rightarrow \varepsilon \approx 0{,}32$

$f(n) = n^2 = O(n^{\log_2 5 - 0{,}32})$

$\Downarrow$

$T(n) = \Theta(n^{\log_2 5})$

g) $T(n) = T(\frac{n}{2}) + 1$

$a = 1 \quad b = 2 \quad f(n) = 1 = n^0$

$\log_2 1 = 0$

$f(n) = n^0 = \Theta(n^{\log_2 1}) = \Theta(n^0) = \Theta(1)$

$\Downarrow$

$T(n) = \Theta(n^{\log_2 1} \log n) = \Theta(n^0 \log n) = \Theta(\log n)$

zad 2

insertion-sort z wartownikiem

```
template <class T>
void insertionsort_ss(T *p, T*k)
{
    T y = p[-1];
    p[-1] = std::numeric_limits<T>::min();
    for(T *s = p+1; s<k; s++)
    {
        T x = *s;
        T *s1 = s;
        while (*(--s1)>x)
            *(s1+1) = *s1;
        *(s1+1) = x;
    }
    p[-1] = y;
}
```

Dwa elementy a_i i a_j w ciągu $A = (a_1 \dots a_n)$ są w inwersji, gdy nie są uporządkowane, czyli $a_i > a_j$ dla $i < j$.

Ile porównań wykona algorytm?

$k + (n-1)$

gdzie, k - liczba inwersji, $n-1$ - należy każdą inwersję zamienić $n-1$ razy, ponieważ jest to liczba porównań z elementami niezamienionymi miejscami

Jaka jest możliwa liczba inwersji dla danych rozmiaru n?

$n-1 + n-2 + n-3 + \dots + 1 = \frac{(n-1)+1}{2} \cdot (n-1) = \frac{n(n-1)}{2}$

W każdej liście może być od 0 (ciąg uporządkowany) do $\frac{n(n-1)}{2}$ (ciąg odwrotnie uporządkowany) inwersji.

liste uporządkowane

przykład + wyjaśnienie

pesymistyczna sytuacja → lista odwrotnie posortowana

4 $\underbrace{3}_{2}$ 2 1 → porównań/inwersji n-1 = 1

4 > 3 → 3 4 $\underbrace{2}_{3}$ 1 → n-1 i porównań 2

4 > 2
3 > 2

łącznie 6 porównań

2 3 4 $\underbrace{1}_{4}$ → n-1 i porównań 3

4 > 1
3 > 1 → 1 2 3 4
2 > 1

n-1 - porównania dla elementu a_n

porównujemy n-1 ilość razy (bo pomijamy pierwszy wyraz)

k - liczba inwersji (liczba takich par)

$\underbrace{\left.\begin{matrix} 4>3 \\ 4>2 \\ 4>1 \end{matrix}\right\} n-1}_{n-1}$ $\underbrace{\left.\begin{matrix} 3>2 \\ 3>1 \end{matrix}\right\} n-2 \quad 2>1 \quad 1}_{k}$ → 6 elementów łącznie

średnia złożoność algorytmu

min = 0 (dla listy posortowanej)

max = $\frac{n(n+1)}{2}$ (dla listy odwrotnie posortowanej)

średnia = $\dfrac{0 + \frac{n(n+1)}{2}}{2} = \dfrac{n(n+1)}{4}$