Семинарни по ДАА

Траян Господинов

20 октомври 2022 г.

Съдържание

Глава 1

Асимптотични нотации

1.1 Теория

Започваме с припомняне на две дефинции от математическия анализ.

Дефиниция 1.1: Асимптотично неотрицателна функция

Ще казваме, че $f : \mathbb{R}^+ \to \mathbb{R}$ е асимптотично неотрицателна, ако $\exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0$ е изпълнено $f(n) \geq 0$

Дефиниция 1.2: Асимптотично положителна функция

Ще казваме, че $f : \mathbb{R}^+ \to \mathbb{R}$ е асимптотично положителна, ако $\exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0$ е изпълнено $f(n) > 0$

В настоящия курс ще се интересуваме от горните два типа функции, тъй като тези функции ще са със семантика на *брой стъпки до приключване изпълнението на дадена програма*, което очевидно няма как да ни върне отрицателен резултат. Също така ще се интересуваме да сравняваме асимптотично неотрицателни/положителни функции. За целта въвеждаме следните пет класа от фунции спрямо дадена функция.

Дефиниция 1.3: Основни класове от функции, спрямо асимптотиката

Нека g е асимптотично неотрицателна функция. Петте основни класа спрямо асимптотиката на g са следните:

- $O(g) = \{f \text{ - неотр.} \mid \exists c > 0\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0 \left(0 \leq f(n) \leq c.g(n)\right)\}$
- $o(g) = \{f \text{ - неотр.} \mid \forall c > 0\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0 \left(0 \leq f(n) \leq c.g(n)\right)\}$
- $\Omega(g) = \{f \text{ - неотр.} \mid \exists c > 0\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0 \left(0 \leq c.g(n) \leq f(n)\right)\}$
- $\omega(g) = \{f \text{ - неотр.} \mid \forall c > 0\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0 \left(0 \leq c.g(n) \leq f(n)\right)\}$
- $\theta(g) = \{f \text{ - неотр.} \mid \exists c_1 > 0\, \exists c_2 > 0\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0 \left(0 \leq c_1.g(n) \leq f(n) \leq c_2.g(n)\right)\}$

Забележка. Поради исторически причини е прието да се пише $f = O(g)$ вместо $f \in O(g)$. Аналогично и за другите четири класа.

Пример 1.1. $O(n^2) = \{n^2, 100n^2, \frac{n^2}{5}, n, 10^6 n, 1, 5n^2 - 1000n - 1000, \dots\}$.
Нека разгледаме при какви стойности на c и n_0 *работи* дефиницията за $g(n) = n^2$ и $f(n)$ съответно:

- $f(n) = n^2$

  Директно се вижда, че $c = 1$ и $n_0 = 0$ ни вършат работа:

  $\forall n \geq \underbrace{0}_{n_0}$ е изпълнено $0 \leq \underbrace{n^2}_{f(n)} \leq \underbrace{1}_{c} * \underbrace{n^2}_{g(n)}$

- $f(n) = 100n^2$

  Отново лесно се преценя, че $c = 100$ и $n_0 = 0$ ни вършат работа:

  $\forall n \geq \underbrace{0}_{n_0}$ е изпълнено $0 \leq \underbrace{100n^2}_{f(n)} \leq \underbrace{100}_{c} * \underbrace{n^2}_{g(n)}$

- $f(n) = \frac{n^2}{5}$

  Отново $c = 1$ и $n_0 = 0$ ни вършат работа (разбира се и $c = \frac{1}{5}$ $n_0 = 0$ също):

  $\forall n \geq \underbrace{0}_{n_0}$ е изпълнено $0 \leq \underbrace{\frac{n^2}{5}}_{f(n)} \leq \underbrace{1}_{c} * \underbrace{n^2}_{g(n)}$

- $f(n) = 5n^2 - 1000n - 1000$

  Макар и сташно на първи поглед, отново лесно се преценя, че $c = 10^7$ и $n_0 = 0$ ни вършат работа (разбира се това не е минимален избор на c и n_0.. интересува ни само да намерим кои да е):

  $\forall n \geq \underbrace{0}_{n_0}$ е изпълнено $0 \leq \underbrace{5n^2 - 1000n - 1000}_{f(n)} \leq \underbrace{10^7}_{c} * \underbrace{n^2}_{g(n)}$

Упражнете се с другите четири класа.

Нотация 1.1: Основни релации над асимптотично неотрицателни функции

Въвеждаме следните пет релации за удобство:

- $f \preccurlyeq g \overset{\text{def}}{\leftrightarrow} f = O(g)$
- $f \prec g \overset{\text{def}}{\leftrightarrow} f = o(g)$
- $f \succcurlyeq g \overset{\text{def}}{\leftrightarrow} f = \Omega(g)$
- $f \succ g \overset{\text{def}}{\leftrightarrow} f = \omega(g)$
- $f \asymp g \overset{\text{def}}{\leftrightarrow} f = \theta(g)$ (асимптотично равенство с точност до константа)

Нотация 1.2: Асимптотично равенство

$f \sim g \overset{\text{def}}{\leftrightarrow} \lim\limits_{n \to \infty} \frac{f(n)}{g(n)} = 1$

Основни свойства: Асимптотични релации

Нека f и g са асимптотично положителни (ще делим на g и затова полож.). Тогава:

1. $f \sigma g \wedge g \sigma h \to f \sigma h$, $\sigma \in \{\preccurlyeq, \prec, \succcurlyeq, \succ, \asymp\}$
2. $f \sigma f$, $\sigma \in \{\preccurlyeq, \succcurlyeq, \asymp\}$
3. $f \preccurlyeq g \wedge g \preccurlyeq f \to f \asymp g$
4. $f \asymp g \leftrightarrow g \asymp f$
5. $f \preccurlyeq g \leftrightarrow g \succcurlyeq f$ (аналогично $f \prec g \leftrightarrow g \succ f$)
6. $f + g \asymp max(f, g)$
   Док: $\frac{f(n)+g(n)}{2} \leq max(f(n), g(n)) \leq f(n) + g(n)$
7. $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = 0 \leftrightarrow f \prec g \, (f = o(g))$
   Док: От деф. на лимит имаме: $\forall \varepsilon > 0 \, \exists n_0 \in \mathbb{N}_0 \, \forall n \geq n_0 : -\varepsilon < \frac{f(n)}{g(n)} < \varepsilon$. Сега умножаваме двете страни по $g(n)$ и получаваме $-g(n) * \varepsilon < 0 < f(n) < g(n) * \varepsilon$, откъдето имаме $\forall \varepsilon > 0 \, \exists n_0 \in \mathbb{N}_0 \, \forall n \geq n_0 : 0 \leq f(n) \leq g(n) * \varepsilon$, което е точно дефиницията на $o(g)$. Обратната посока е аналогична.
8. $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = c > 0 \to f \asymp g \, (f = \theta(g))$
   Док: От деф. на лимит имаме: $\forall \varepsilon > 0 \, \exists n_0 \in \mathbb{N}_0 \, \forall n \geq n_0 : c - \varepsilon < \frac{f(n)}{g(n)} < c + \varepsilon$. Сега умножаваме двете страни по $g(n)$ и получаваме $g(n)*(c-\varepsilon) < f(n) < g(n)*(c+\varepsilon)$. Тъй като $c > 0$, то е ясно че има $\varepsilon > 0 : c - \varepsilon > 0$. Тоест получихме следното: $\exists \varepsilon > 0 \, \exists n_0 \in \mathbb{N}_0 \, \forall n \geq n_0 : 0 < g(n) * \underbrace{(c-\varepsilon)}_{c_1} < f(n) < g(n) * \underbrace{(c+\varepsilon)}_{c_2}$, което е точно дефиницията на $\theta(g)$.
   Обратната посока **не** е вярна.
9. Нека g не е ограничена отгоре и нека $a > 1$. Тогава:
   (а) $f \prec g \to a^{f(n)} \prec a^{g(n)}$ (нестрогия аналог **не** е верен)
   (б) $log_a f(n) \prec log_a g(n) \to f(n) \prec g(n)$ (нестрогия аналог **не** е верен)
10. $\forall a > 1 \, \forall t > 0 \, \forall \varepsilon > 0 \left(log_a^t(n) \prec n^\varepsilon \right)$

Факт 1.1: Апроксимация на Стирлинг

$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n (1 + \frac{1}{12n} + \frac{1}{288n^2} + \dots)$

Следствие 1.1: Асимптотична апроксимация на Стирлинг

$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$

Приложение.

- $log(n!) \asymp nlog(n)$ (докажете за упражнение)
- $\binom{2n}{n} = \frac{(2n)!}{n!n!} \sim \frac{\sqrt{4\pi n}\left(\frac{2n}{e}\right)^{2n}}{2\pi n\left(\frac{n}{e}\right)^{2n}} = \frac{2^{2n}n^{2n}}{\sqrt{\pi n}\, n^{2n}} = \frac{2^{2n}}{\sqrt{\pi n}} = \frac{4^n}{\sqrt{\pi n}}$

Факт 1.2: Логаритми и техните свойства

$\forall a \in \mathbb{R}^+ \backslash \{1\} \forall b \in \mathbb{R}^+ : a^x = b \leftrightarrow x = log_a(b)$

1. $a^{log_a(b)} = b$
2. $log_{a^n}(b^m) = \frac{m}{n} log_a(b)$
3. (а) $log_a(x) + log_a(y) = log_a(xy)$
   (б) $log_a(x) - log_a(y) = log_a(\frac{x}{y})$
4. $log_a(x) = \frac{log_b(x)}{log_b(a)}$, $b \in \mathbb{R}^+ \backslash \{1\}$
   (а) $log_a(b) = \frac{1}{log_b(a)}$
   (б) $log_b(a) * log_a(x) = log_b(x)$
5. $a^{log_b(x)} = x^{log_b(a)}$

1.2 Задачи

Задача 1.1. Нека $p(x) = a_0x^k + \cdots + a_k$ е асимптотично положителен полином (т.е. $a_0 > 0$). Да се докаже, че $p(n) \asymp n^k$.

Решение. $\lim\limits_{n\to\infty} \frac{p(n)}{n^k} = \lim\limits_{n\to\infty} \frac{a_0x^k+\cdots+a_k}{n^k} = \underbrace{\lim\limits_{n\to\infty} \frac{a_0n^k}{n^k}}_{a_0} + \underbrace{\lim\limits_{n\to\infty} \frac{a_1n^{k-1}}{n^k}}_{0} + \cdots + \underbrace{\lim\limits_{n\to\infty} \frac{a_n}{n^k}}_{0} = a_0 > 0.$
От основно свойство 8 следва, че $p(n) \asymp n^k$.

Задача 1.2. Нека $k \in \mathbb{N}^+$. Да се докаже, че $\binom{n}{k} \asymp n^k$.

Решение. $\lim\limits_{n\to\infty} \frac{\binom{n}{k}}{n^k} = \lim\limits_{n\to\infty} \frac{n(n-1)\ldots(n-k+1)}{n^k k!} = \frac{1}{k!} > 0$. От основно свойство 8 следва, че $\binom{n}{k} \asymp n^k$.

Задача 1.3. Да се докаже, че $(n+1)^n \asymp n^n$.

Решение. $\lim\limits_{n\to\infty} \frac{(n+1)^n}{n^n} = \lim\limits_{n\to\infty} \left(\frac{n+1}{n}\right)^n = e > 0$. От основно свойство 8 следва, че $(n+1)^n \asymp n^n$.

Задача 1.4. Нека $f(n) = \begin{cases} 1 & , \lfloor n \rfloor \equiv 0(2) \\ n^2 & , \lfloor n \rfloor \equiv 1(2) \end{cases}$ и $g(n) = n$. Да се докаже, че те са асимптотично несравними.

Решение. За упражнение...

Задача 1.5. Вярна ли е следната импликация: $f = O(g) \to (f = o(g) \vee f = \theta(g))$?

Решение. Не е вярна! Проверете за упражнение със следния контрапример:
$f(n) = \begin{cases} n & , \lfloor n \rfloor \equiv 0(2) \\ 1/n & , \lfloor n \rfloor \equiv 1(2) \end{cases}$ и $g(n) = n$.

Задача 1.6. Да се сортират по асимптотика следните функции:

$f_1(n) = n^3$	$f_2(n) = \sqrt{n}$	$f_3(n) = log(n)$	$f_4(n) = log^2(n)$
$f_5(n) = log^{(2)}(n)$	$f_6(n) = n!$	$f_7(n) = a^n$	$f_8(n) = a$
$f_9(n) = n^n$	$f_{10}(n) = n^{-2}$	$f_{11}(n) = n^2$	$f_{12}(n) = n^{log(n)}$

Решение.
$n^n \succ n! \succ a^n \succ n^{log(n)} \succ n^3 \succ n^2 \succ \sqrt{n} \succ log^2(n) \succ log(n) \succ log^{(2)}(n) \succ a \succ n^{-2}$
Ще опишем подробно решенията:

1. $\boldsymbol{n^n \succ n!}$

   От анализа знаем, че $\lim\limits_{n\to\infty} \frac{n!}{n^n} = 0$. Тогава от основно свойство 7 следва, че $n! \prec n^n$.

2. $\boldsymbol{n! \succ a^n}$

   Ще приложим основно свойство 9 ... т.е. логаритмуваме двете страни и получаваме

   $$log(n!) \,?\, log(a^n)$$
   $$nlog(n) \,?\, nlog(a)$$

   Знаем, че $\lim\limits_{n\to\infty} \frac{nlog(a)}{nlog(n)} = 0$. Тогава от основно свойство 7 следва, че $nlog(a) \prec nlog(n)$. Оттук и от основно свойство 9 следва, че $a^n \prec n!$.

3. $\boldsymbol{a^n \succ n^{log(n)}}$

   Ще приложим основно свойство 9 ... т.е. логаритмуваме двете страни и получаваме

   $$log(a^n) \,?\, log(n^{log(n)})$$
   $$nlog(a) \,?\, log(n)log(n)$$

   От основно свойство 10 имаме $log(n)log(n) \prec nlog(a)$. Оттук и от основно свойство 9 следва, че $n^{log(n)} \prec a^n$.

4. $\boldsymbol{n^{log(n)} \succ n^3}$

   Ще приложим основно свойство 9 ... т.е. логаритмуваме двете страни и получаваме

   $$log(n^{log(n)}) \,?\, log(n^3)$$
   $$log^2(n) \,?\, 3log(n)$$

   Знаем, че $\lim\limits_{n\to\infty} \frac{3log(n)}{log^2(n)} = 0$. Тогава от основно свойство 7 следва, че $3log(n) \prec log^2(n)$. Оттук и от основно свойство 9 следва, че $n^3 \prec n^{log(n)}$.

5. $\boldsymbol{n^3 \succ n^2}$

Знаем, че $\lim\limits_{n\to\infty} \frac{n^2}{n^3} = 0$. Тогава от основно свойство 7 следва, че $n^2 \prec n^3$.

6. $\boldsymbol{n^2 \succ \sqrt{n}}$

Знаем, че $\lim\limits_{n\to\infty} \frac{\sqrt{n}}{n^2} = 0$. Тогава от основно свойство 7 следва, че $\sqrt{n} \prec n^2$.

7. $\boldsymbol{\sqrt{n} \succ log^2(n)}$

От основно свойство 9 директно следва, че $log^2(n) \prec \sqrt{n}$.

8. $\boldsymbol{log^2(n) \succ log(n)}$

Знаем, че $\lim\limits_{n\to\infty} \frac{log(n)}{log^2(n)} = 0$. Тогава от основно свойство 7 следва, че $log(n) \prec log^2(n)$.

9. $\boldsymbol{log(n) \succ log(log(n))}$

Полагаме $m = log(n)$ и получаваме

$$m \,?\, log(m)$$

От основно свойство 10 следва, че $log(m) \prec m$. Сега като върнем полагането получаваме $log(log(n)) \prec log(n)$.

10. $\boldsymbol{log(log(n)) \succ a}$

Знаем, че $\lim\limits_{n\to\infty} \frac{a}{log^{(2)}(n)} = 0$. Тогава от основно свойство 7 следва, че $a \prec log^{(2)}(n)$.

11. $\boldsymbol{a \succ n^{-2}}$

Знаем, че $\lim\limits_{n\to\infty} \frac{n^{-2}}{a} = 0$. Тогава от основно свойство 7 следва, че $n^{-2} \prec a$.

Задача 1.7. Да се сортират по асимптотика следните функции:

$f_1(n) = (\sqrt{2})^{log(n)}$	$f_2(n) = n^3$	$f_3(n) = n!$	$f_4(n) = (log(n))!$
$f_5(n) = e^{-2ln(n)}$	$f_6(n) = log^2(n)$	$f_7(n) = log(n!)$	$f_8(n) = 2^{2^n}$
$f_9(n) = n^{\frac{1}{log(n)}}$	$f_{10}(n) = log^{(2)}(n)$	$f_{11}(n) = \left(\frac{3}{2}\right)^n$	$f_{12}(n) = n2^n$
$f_{13}(n) = 4^{log(n)}$	$f_{14}(n) = (n+1)!$	$f_{15}(n) = \sqrt{log(n)}$	$f_{16}(n) = 2^{\sqrt{2log(n)}}$
$f_{17}(n) = n^{log(log(n))}$	$f_{18}(n) = log(n)$	$f_{19}(n) = 2^{log(n)}$	$f_{20}(n) = (log(n))^{log(n)}$

Решение.
$e^{-2ln(n)} \prec n^{\frac{1}{log(n)}} \prec log^{(2)}(n) \prec \sqrt{log(n)} \prec log(n) \prec log^2(n) \prec 2^{\sqrt{2log(n)}} \prec (\sqrt{2})^{log(n)} \prec 2^{log(n)} \prec log(n!) \prec 4^{log(n)} \prec n^3 \prec (log(n))! \prec (log(n))^{log(n)} \asymp n^{log(log(n))} \prec \left(\frac{3}{2}\right)^n \prec n2^n \prec n! \prec (n+1)! \prec 2^{2^n}$
Ще опишем подробно решенията:

1. $\boldsymbol{e^{-2ln(n)} \prec n^{\frac{1}{log(n)}}}$

Ще преобразуваме функциите използвайки основните свойства на логаритмите:

$$(e^{ln(n)})^{-2} \,?\, n^{log_n(2)}$$

$$(n^{ln(e)})^{-2} \,?\, 2^{log_n(n)}$$

$$n^{-2} \,?\, 2$$

Сега знаем, че $\lim\limits_{n\to\infty} \frac{n^{-2}}{2} = 0$. Тогава от основно свойство 7 следва, че $n^{-2} \prec 2$.

2. $\boldsymbol{n^{\frac{1}{log(n)}} \prec log^{(2)}(n)}$

Вече показахме, че $n^{\frac{1}{log(n)}} = 2$. Сега знаем, че $\lim\limits_{n\to\infty} \frac{2}{log^{(2)}(n)} = 0$. Тогава от основно свойство 7 следва, че $2 \prec log^{(2)}(n)$.

3. $\boldsymbol{log^{(2)}(n) \prec \sqrt{log(n)}}$

Полагаме $m = log(n)$ и получаваме

$$log(m) \,?\, \sqrt{m}$$

От основно свойство 10 следва, че $log(m) \prec m$. Сега като върнем полагането получаваме $log^{(2)}(n) \prec \sqrt{log(n)}$.

4. $\boldsymbol{\sqrt{log(n)} \prec log(n)}$

Директно прилагаме основно свойство 7.

5. $\boldsymbol{log(n) \prec log^2(n)}$

Директно прилагаме основно свойство 7.

6. $\boldsymbol{(log(n))^2 \prec 2^{\sqrt{2log(n)}}}$

Ще приложим основно свойство 9 . . . т.е. логаритмуваме двете страни и получаваме

$$log((log(n))^2) \,?\, log(2^{\sqrt{2log(n)}})$$

$$2log(log(n)) \,?\, \sqrt{2log(n)}log(2)$$

Полагаме $m = log(n)$ и получаваме

$$2log(m) \,?\, \sqrt{2m}$$

От основно свойство 10 имаме $2log(m) \prec \sqrt{2m}$. Сега като върнем полагането имаме $2log(log(n)) \prec \sqrt{2log(n)}$. Оттук и от основно свойство 9 следва, че $(log(n))^2 \prec 2^{\sqrt{2log(n)}}$.

7. $\boldsymbol{2^{\sqrt{2log(n)}} \prec (\sqrt{2})^{log(n)}}$

Ще приложим основно свойство 9 . . . т.е. логаритмуваме двете страни и получаваме

$$log(2^{\sqrt{2log(n)}}) \,?\, log(\sqrt{n})$$

$$\sqrt{2log(n)} \,?\, \frac{1}{2}log(n)$$

Знаем, че $\lim_{n\to\infty} \frac{\sqrt{2log(n)}}{\frac{1}{2}log(n)} = 0$. Тогава от основно свойство 7 следва, че $\sqrt{2log(n)} \prec \frac{1}{2}log(n)$. Оттук и от основно свойство 9 следва, че $2^{\sqrt{2log(n)}} \prec (\sqrt{2})^{log(n)}$.

8. $\boldsymbol{(\sqrt{2})^{log(n)} \prec 2^{log(n)}}$

Ще преобразуваме функциите използвайки основните свойства на логаритмите:

$$n^{log(\sqrt{2})} \,?\, n^{log(2)}$$

$$\sqrt{n} \,?\, n$$

Сега знаем, че $\lim_{n\to\infty} \frac{\sqrt{n}}{n} = 0$. Тогава от основно свойство 7 следва, че $\sqrt{n} \prec n$.

9. $\boldsymbol{2^{log(n)} \prec log(n!)}$

Вече показахме, че $2^{log(n)} = n$ и $log(n!) \asymp nlog(n)$. Сега знаем, че $\lim_{n\to\infty} \frac{n}{nlog(n)} = 0$. Тогава от основно свойство 7 следва, че $n \prec nlog(n)$.

10. $\boldsymbol{log(n!) \prec 4^{log(n)}}$

Ще преобразуваме $4^{log(n)}$ използвайки основните свойства на логаритмите:

$$log(n!) \,?\, n^{log(4)}$$

$$nlog(n) \,?\, n^2$$

Тогава от основно свойство 10 следва, че $nlog(n) \prec n^2$.

11. $\boldsymbol{4^{log(n)} \prec n^3}$

Вече показахме, че $4^{log(n)} = n^2$. Сега знаем, че $\lim_{n\to\infty} \frac{n^2}{n^3} = 0$. Тогава от основно свойство 7 следва, че $n^2 \prec n^3$.

12. $\boldsymbol{n^3 \prec (log(n))!}$

Ще преобразуваме $(log(n))!$ използвайки апроксимацията на Стирлинг:

$$(log(n))! \sim \sqrt{2\pi log(n)}\left(\frac{log(n)}{e}\right)^{log(n)}$$

Ще приложим основно свойство 9 . . . т.е. логаритмуваме двете страни и получаваме

$$log(n^3) \,?\, log\left(\sqrt{2\pi log(n)}\left(\frac{log(n)}{e}\right)^{log(n)}\right)$$

$$3log(n) \,?\, \underbrace{log\left(\sqrt{2\pi log(n)}\right)}_{\asymp log(log(n))} + log(n)\,log(log(n)) - log(n)log(e)$$

$$log(n) \,?\, log(n)\,log(log(n))$$

Знаем, че $\lim_{n\to\infty} \frac{log(n)}{log(n)\,log(log(n))} = 0$. Тогава от основно свойство 7 следва, че $log(n) \prec log(n)\,log(log(n))$. Оттук и от основно свойство 9 следва, че $n^3 \prec (log(n))!$.

13. $\boldsymbol{(log(n))! \prec (log(n))^{log(n)}}$

Полагаме $m = log(n)$ и получаваме

$$m! \,?\, m^m$$

От анализа знаем, че $\lim_{n\to\infty} \frac{m!}{m^m} = 0$. Тогава от основно свойство 7 следва, че $m! \prec m^m$. Сега като върнем полагането получаваме $(log(n))! \prec (log(n))^{log(n)}$.

14. $\boldsymbol{(log(n))^{log(n)} \asymp n^{log(log(n))}}$

Директно прилагаме основно свойство 5 на логаритмите.

15. $\boldsymbol{n^{log(log(n))} \prec \left(\frac{3}{2}\right)^n}$

Ще приложим основно свойство 9 ... т.е. логаритмуваме двете страни и получаваме

$$log(n)\, log(log(n)) \,?\, n$$

От основно свойство 10 следва, че $log(n)\, log(log(n)) \prec log^2(n) \prec n$. Оттук и от основно свойство 9 следва, че $n^{log(log(n))} \prec \left(\frac{3}{2}\right)^n$.

16. $\boldsymbol{\left(\frac{3}{2}\right)^n \prec n2^n}$

Директно прилагаме основно свойство 7. $//\left(\frac{3}{2}\right)^n \prec 2^n \prec n2^n$

17. $\boldsymbol{n2^n \prec n!}$

Ще приложим основно свойство 9 ... т.е. логаритмуваме двете страни и получаваме

$$log(n2^n) \,?\, log(n!)$$

$$log(n) + nlog(2) \,?\, nlog(n)$$

Знаем, че $\lim_{n\to\infty} \frac{n}{nlog(n)} = 0$. Тогава от основно свойство 7 следва, че $n \prec log(n)$. Оттук и от основно свойство 9 следва, че $n2^n \prec n!$.

18. $\boldsymbol{n! \prec (n+1)!}$

Директно прилагаме основно свойство 7

19. $\boldsymbol{(n+1)! \prec 2^{2^n}}$

Ще приложим основно свойство 9 ... т.е. логаритмуваме двете страни и получаваме

$$(n+1)log(n+1) \,?\, 2^n$$

Логаритмуваме още веднъж

$$log(n+1)\, log(log(n+1)) \,?\, n$$

От основно свойство 10 следва, че $log(n)\, log(log(n)) \prec log^2(n) \prec n$. Оттук и от основно свойство 9 следва, че $log(n)\, log(log(n)) \prec 2^n$. И отново от основно свойство 9 следва, че $(n+1)! \prec 2^{2^n}$.

Глава 2

Анализ на коректността на алгоритми

2.1 Итеративни алгоритми

Коректност на итеративни алгоритми ще доказваме чрез ***инвариант***. Това е пособие, което много наподобява доказателство чрез индукция. Аналогично на индукцията имаме ***база***. Индуктивната хипотеза и индуктивната стъпка тук са обединени в една фаза - ***поддръжка***. Имаме нова фаза наречена ***терминация***.

Забележка 2.1

Важно е да отбележим, че инварианта е пособие за доказателство за коректност на итеративен цикъл, а **не** за цял алгоритъм!

Задача 2.1. Даден е следния алгоритъм (зададен чрез ***псевдокод***):

```
1. Alg1(n) : // n ∈ ℕ_0
2.     s ← 1;
3.     for i ← 1 to n
4.         s ← s * 2;
5.     return s;
```

Какво връща той? Да се докаже формално.

Решение. Ще докажем, че $Alg1(n) = 2^n$.

Инвариант

При всяко k-то достигане на ред 3 имаме, че $s = 2^{k-1}$ (k броим от 1)

База. При $k = 1$-во достигане на ред 3 имаме, че $s \overset{\text{ред } 3}{=} 1 = 2^0 = 2^{k-1}$.

Поддръжка. Нека е вярно за някое k-то непоследно достигане на ред 3, т.е. имаме $s = 2^{k-1}$. Сега се изпълнява тялото на цикъла, т.е. $s_{new} \overset{\text{ред } 4}{=} s_{old} * 2 \overset{\text{ИП}}{=} 2^{k-1} * 2 = 2^k = 2^{(k+1)-1}$. След изпълнение на тялото на цикъла отново се връщаме на ред 3 като вече го достигаме за $(k+1)$-ви път (може да е последно достигане) и имаме $s = 2^{(k+1)-1}$.

Терминация. При $k = (n+1)$-то достигане на ред 3 тялото на цикъла няма да се изпълни. От инварианта имаме, че $s = 2^{(n+1)-1} = 2^n$. Директно връщаме $s = 2^n$ на ред 5.

Задача 2.2. Даден е следния алгоритъм:

```
1. Alg2(A[1..n]) : // n ∈ ℕ⁺ , A ∈ ℝⁿ
2.     s ← 0;
3.     for i ← 1 to n
4.         s ← s + A[i];
5.     return s;
```

Какво връща той? Да се докаже формално.

Решение. Ще докажем, че $Alg2(A[1..n]) = \sum_{i=1}^{n} A[i]$.

Инвариант

При всяко k-то достигане на ред 3 имаме:

- $i = k$ (в рамките на курса ще го считаме за тривиално при *стандартни* for-цикли и няма да го доказваме)
- $s = \sum_{i=1}^{k-1} A[i]$

База. При $k = 1$-во достигане на ред 3 имаме, че $s \overset{\text{ред } 2}{=} 0 = \sum_{i=1}^{0} A[i] = \sum_{i=1}^{k-1} A[i]$.

Поддръжка. Нека е вярно за някое непоследно достигане на ред 3, т.е. имаме $s = \sum_{i=1}^{k-1} A[i]$.

Сега се изпълнява тялото на цикъла $s_{new} \overset{\text{ред } 4}{=} s_{old} + A[i] = s_{old} + A[k] \overset{\text{ИП}}{=} \sum_{i=1}^{k-1} A[i] + A[k] = \sum_{i=1}^{k} A[i]$.

Терминация. При $k = (n+1)$-то достигане на ред 3 тялото на цикъла няма да се изпълни. От инварианта имаме, че $s = \sum_{i=1}^{n} A[i]$. Директно връщаме s на ред 5.

2.2 Рекурсивни алгоритми

За разлика от итеративните алгоритми, коректност на рекурсивни алгоритми ще доказваме чрез добре познатата ни ***пълна математическа индукция*** състояща се от три фази - ***база***, ***индуктивна хипотеза*** и ***индуктивна стъпка***.

Забележка 2.2

Индукцията е пособие за доказателство за коректност на цял алгоритъм!

Забележка 2.3: Рекурсивен алгоритъм, съдържащ итеративен цикъл

За да докажем рекурсивен алгоритъм, съдържащ итеративен (един или повече) цикъл, то трябва да направим доказателство за коректност на всеки итеративен цикъл (използвайки ***инвариант***) след което да използваме ***пълна математическа индукция*** за доказателство за коректност на целия алгоритъм.

Задача 2.3. Даден е следния алгоритъм:

```
1. Alg3(a, n) : // a ∈ ℝ, n ∈ ℕ_0
2.   if n = 0 then
3.     return 1;
4.   if n ≡ 0 (mod 2) then
5.     return Alg3(a * a, n/2);
6.   return a * Alg3(a, n − 1);
```

Какво връща той? Да се докаже формално.

Решение. Ще докажем, че $Alg3(a,n) = \begin{cases} a^n & , a \neq 0 \wedge n \neq 0 \\ 1 & , a = 0 \vee n = 0 \end{cases}$ чрез пълна математическа индукция по n.

База. $(n = 0)$
Разглеждаме два случая:

сл.1 $(a = 0)$ - $Alg3(a,0) \overset{\text{2-3}}{\underset{\text{ред}}{=}} 1$.

сл.2 $(a \neq 0)$ - $Alg3(a,0) \overset{\text{2-3}}{\underset{\text{ред}}{=}} 1$.

Индуктивна хипотеза. Нека $\forall m \leq n$ е изпълнено, че $Alg3(a,m) = \begin{cases} a^m & , a \neq 0 \wedge m \neq 0 \\ 1 & , a = 0 \vee m = 0 \end{cases}$

Индуктивна стъпка. Ще докажем за $n+1 > 0$. Отново разглеждаме два случая:

сл.1 $(n+1) \equiv 0 \,(mod\, 2)$
Тъй като $n+1 > 0$, то не влизаме в тялото на if на ред 2-3. Знаем, че $n+1 > 0$ и $(n+1) \equiv 0 \,(mod\, 2)$, значи имаме $n+1 \geq 2$. Откъдето заключваме, че $\frac{n+1}{2} > 0$. Влизаме в тялото на if на ред 4-5, като директно връщаме $Alg3(a*a, \underbrace{\frac{n+1}{2}}_{>0}) \overset{\text{ИХ}}{=} (a*a)^{\frac{n+1}{2}} = a^{n+1}$.

сл.2 $(n+1) \equiv 1 \,(mod\, 2)$
Тъй като $n+1 > 0$, то не влизаме в тялото на if на ред 2-3. От друга страна не влизаме в тялото на if на ред 4-5 заради $(n+1) \equiv 1 \,(mod\, 2)$. Тоест достигаме ред 6, където директно връщаме $a*Alg3(a,(n+1)-1) = a*Alg3(a,n) \overset{\text{ИХ}}{=} a* \begin{cases} a^n & , a \neq 0 \wedge n \neq 0 \\ 1 & , a = 0 \vee n = 0 \end{cases} = a^{n+1}$ (отново е важно, че $n+1 > 0$.. т.е. не е нужно да разглеждаме два случая).

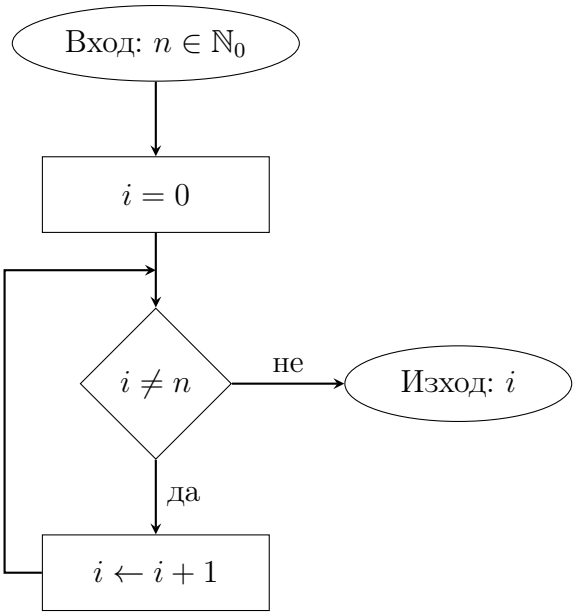
2.3 Блок схеми

В Забележка 2.1 споменахме, че ***инвариант*** ни служи за доказване само на итеративни цикли, а не на цели алгоритми, а в Забележка 2.2 казахме, че използваме ***индукция*** за доказване на коректността на цели алгоритми. Това поражда въпроса: "С какво пособие доказваме итеративни алгоритми?". Отговора: чрез ***блок схеми***.

Забележка 2.4

Разглежданите итеративни алгоритми в настоящия курс са достатъчно прости, т.е. целия алгоритъм всъщност е само един (най-много два) цикъл(а). За тези алгоритми е напълно приемливо да се използва ***инвариант*** за доказателство за коректност на целия алгоритъм, но когато алгоритъма не е просто един цикъл, то тогава се налага да доказваме коректност чрез ***блок схеми***. Поради тази причина в рамките на текущия курс не се изискват познания за ***блок схеми***.

Задача 2.4. Даден е следния алгоритъм (зададен чрез ***блок схема***):



Какво връща той? Да се докаже формално.

Решение. Ще докажем, че алгоритъма връща n.

Дефинираме три предиката (по един за всяка *не правоъгълна кутийка*):

- $I(n) \overset{def}{\leftrightarrow} n \in \mathbb{N}_0$
- $L(n, i) \overset{def}{\leftrightarrow} (n \in \mathbb{N}_0 \wedge i \in \mathbb{N}_0)$
- $O(n, i) \overset{def}{\leftrightarrow} i = n$

Ще докажем следните три импликации (връзките между *не правоъгълните кутийки*):

- $(I(n) \wedge i = 0) \rightarrow L(n, i)$
- $(L(n, i) \wedge i \neq n) \rightarrow L(n, i + 1)$
- $(L(n, i) \wedge \neg(i \neq n)) \rightarrow O(n, i)$

Ето и пълното доказателство на горните три импликации:

- $(I(n) \wedge i = 0) \rightarrow L(n, i)$
  Нека са изпълнени $\underbrace{n \in \mathbb{N}_0}_{I(n)}$ и $i = 0$. Чудим се дали е изпълнено $\underbrace{n \in \mathbb{N}_0 \text{ и } i \in \mathbb{N}_0}_{L(n,i)}$.
  Очевидно е изпълнено.
- $(L(n, i) \wedge i \neq n) \rightarrow L(n, i + 1)$
  Нека са изпълнени $\underbrace{n \in \mathbb{N}_0 \text{ и } i \in \mathbb{N}_0}_{L(n,i)}$ и $i \neq n$. Чудим се дали е изпълнено $\underbrace{n \in \mathbb{N}_0 \text{ и } (i + 1) \in \mathbb{N}_0}_{L(n,i)}$.
  Очевидно е изпълнено.
- $(L(n, i) \wedge \neg(i \neq n)) \rightarrow O(n, i)$
  Нека са изпълнени $\underbrace{n \in \mathbb{N}_0 \text{ и } i \in \mathbb{N}_0}_{L(n,i)}$ и $\underbrace{\neg(i \neq n)}_{\text{т.е. } i=n}$. Чудим се дали е изпълнено $\underbrace{i = n}_{O(n,i)}$.
  Очевидно е изпълнено.

Тоест доказахме, че достигайки *кутийката за изход* е изпълнен предиката $\underbrace{i = n}_{O(n,i)}$. Тоест връщания резултат наистина е n.

2.4 Често срещани грешки

2.4.1 Бъркане на позицията на инварианта при do-while цикъл

Без да се замисли човек е лесно да се направи тази грешка от невнимание. Да разгледаме следния псевдокод за илюстрация:

```
1. Alg4(n) : // n ∈ ℕ₀
2.     i ← 0;
3.     do
4.         i ← i + 1;
5.     while i < n;
6.     return n;
```

Инвариант: ГРЕШЕН

При всяко k-то достигане на ред **3**...

Инвариант: КОРЕКТЕН

При всяко k-то достигане на ред **5**...

Съобразете защо!

2.4.2 Некоректна терминация

За разлика от предходната грешка, тази е значително по-трудна за съобразяване. Да разгледаме следния алгоритъм:

```
1. Alg5(n) : // n ∈ ℕ_0
2.     i ← 0;
3.     while i < n do
4.         i ← i + 1;
5.     return n;
```

Ясно е, че $Alg5(n) = n$. Проблем настъпва, когато се опитаме да съставим инвариант. Първоначалната ни идея би била да направим следния инвариант:

Инвариант: НЕПЪЛЕН

При всяко k-то достигане на ред 3 имаме, че $i = k - 1$

Този инвариант е абсолютно верен. Нека го докажем.

База. При $k = 1$-во достигане на ред 3 имаме, че $i = 0 = 1 - 1 = k - 1$.

Поддръжка. Нека е изпълнено за някое k-то непоследно достигане на ред 3. Тъй като е непоследно достигане, то тялото на цикъла се изпълнява и достигаме отново ред 3, като имаме $i_{new} \overset{\text{ред 4}}{=} i_{old} + 1 \overset{\text{ИХ}}{=} (k-1) + 1 = k = (k+1) - 1$.

Терминация. При $(n+1)$-вото достигане на ред 3 за първи път не влизаме в тялото на while-а. От инварианта знаем, че $i = (n+1) - 1 = n$, което и връщаме директно след цикъла.

Забелязахте ли проблема?

Проблема е, че не знаем кое е де факто **последното** достигане на ред 3. При текущия алгоритъм е очевидно за съобразяване, но аналогичен проблем може да възникне при много по-сложни алгоритми. Инварианта е абсолютно коректно доказан (съобразете защо поддръжката е коректно доказана), но терминацията е грешна! Ето как би изглеждал правилен инвариант за алгоритъма:

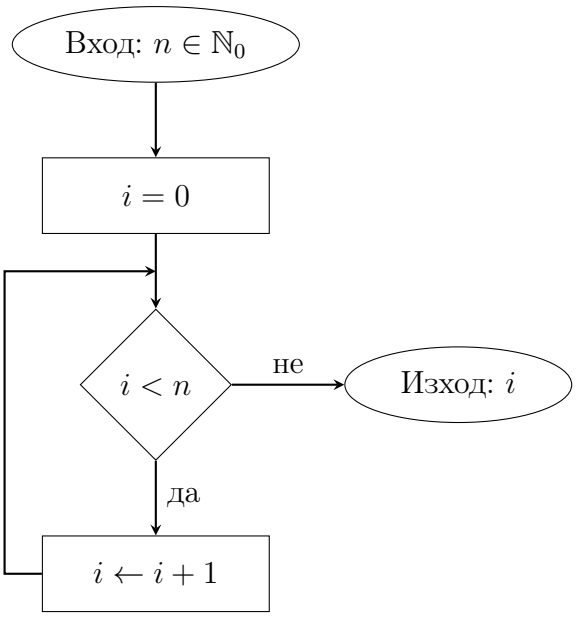
Инвариант

При всяко k-то достигане на ред 3 имаме:

- $i = k - 1$
- $i < n + 1$

Докажете, използвайки този инвариант.

Тъй като е трудно да се съобрази точно къде е проблема, нека да разгледаме същия проблем, но зададен чрез блок схема:



Нека да разгледаме грешната идея първо:

Дефинираме три предиката (по един за всяка *не правоъгълна кутийка*):

- $I(n) \overset{def}{\leftrightarrow} n \in \mathbb{N}_0$
- $L(n,i) \overset{def}{\leftrightarrow} (n \in \mathbb{N}_0 \wedge i \in \mathbb{N}_0)$
- $O(n,i) \overset{def}{\leftrightarrow} i = n$

Ще докажем следните три импликации (връзките между *не правоъгълните кутийки*):

- $(I(n) \wedge i = 0) \rightarrow L(n,i)$
- $(L(n,i) \wedge i < n) \rightarrow L(n, i+1)$
- $(L(n,i) \wedge \neg(i < n)) \rightarrow O(n,i)$

Ето и опит за пълно доказателство на горните три импликации:

- $(I(n) \wedge i = 0) \rightarrow L(n,i)$
  Нека са изпълнени $\underbrace{n \in \mathbb{N}_0}_{I(n)}$ и $i = 0$. Чудим се дали е изпълнено $\underbrace{n \in \mathbb{N}_0 \text{ и } i \in \mathbb{N}_0}_{L(n,i)}$.
  Очевидно е изпълнено.
- $(L(n,i) \wedge i < n) \rightarrow L(n, i+1)$
  Нека са изпълнени $\underbrace{n \in \mathbb{N}_0 \text{ и } i \in \mathbb{N}_0}_{L(n,i)}$ и $i < n$. Чудим се дали е изпълнено $\underbrace{n \in \mathbb{N}_0 \text{ и } (i+1) \in \mathbb{N}_0}_{L(n,i)}$.
  Очевидно е изпълнено.
- $(L(n,i) \wedge \neg(i < n)) \rightarrow O(n,i)$
  Нека са изпълнени $\underbrace{n \in \mathbb{N}_0 \text{ и } i \in \mathbb{N}_0}_{L(n,i)}$ и $\underbrace{\neg(i < n)}_{\text{т.е. } i \geq n}$. Чудим се дали е изпълнено $\underbrace{i = n}_{O(n,i)}$. Не знаем! Единствено имаме, че $i \geq n$ (и че е естествено число), но не знаем дали е n или $n+1$ или дори може да е $n + 10000$.

Тоест не се получи, използвайки тези 3 предиката.. ще трябва да опитаме нещо по-силно.

Дефинираме три предиката (по един за всяка *не правоъгълна кутийка*):

- $I(n) \overset{def}{\leftrightarrow} n \in \mathbb{N}_0$
- $L(n,i) \overset{def}{\leftrightarrow} (n \in \mathbb{N}_0 \wedge i \in \mathbb{N}_0 \wedge \boldsymbol{i \leq n})$
- $O(n,i) \overset{def}{\leftrightarrow} i = n$

Ще докажем следните три импликации (връзките между *не правоъгълните кутийки*):

- $(I(n) \wedge i = 0) \to L(n,i)$
- $(L(n,i) \wedge i < n) \to L(n, i+1)$
- $(L(n,i) \wedge \neg(i < n)) \to O(n,i)$

Ето и пълното доказателство на горните три импликации:

- $(I(n) \wedge i = 0) \to L(n,i)$
  Нека са изпълнени $\underbrace{n \in \mathbb{N}_0}_{I(n)}$ и $i = 0$. Чудим се дали е изпълнено $\underbrace{n \in \mathbb{N}_0 \text{ и } i \in \mathbb{N}_0 \text{ и } i \leq n}_{L(n,i)}$.
  Тъй като $n \in \mathbb{N}_0$, то знаем, че $n \geq 0$, но $i = 0$, тоест имаме $n \geq i$.
- $(L(n,i) \wedge i < n) \to L(n, i+1)$
  Нека са изпълнени $\underbrace{n \in \mathbb{N}_0 \text{ и } i \in \mathbb{N}_0 \text{ и } i \leq n}_{L(n,i)}$ и $i < n$. Чудим се дали е изпълнено следното
  $\underbrace{n \in \mathbb{N}_0 \text{ и } (i+1) \in \mathbb{N}_0 \text{ и } (i+1) \leq n}_{L(n,i)}$. Тъй като имаме, че $i < n$ (и $i \in \mathbb{N}_0$), то е еквивалентно с $(i+1) \leq n$. Очевидно е, че $(i+1) \in \mathbb{N}_0$.
- $(L(n,i) \wedge \neg(i < n)) \to O(n,i)$
  Нека са изпълнени $\underbrace{n \in \mathbb{N}_0 \text{ и } i \in \mathbb{N}_0 \text{ и } i \leq n}_{L(n,i)}$ и $\underbrace{\neg(i < n)}_{\text{т.е. } i \geq n}$. Чудим се дали е изпълнено $\underbrace{i = n}_{O(n,i)}$.
  Очевидно е, тъй като имаме $i \geq n$ и $i \leq n$, то е ясно, че $i = n$.

Доказахме, че алгоритъма връща n (при вход n).

2.5 Задачи

Задача 2.5. Даден е следния алгоритъм:

```
1.  Kadane(A[1..n]) : // n ∈ ℕ⁺ , A ∈ ℤⁿ
2.      c ← A[1];
3.      m ← A[1];
4.      for i ← 2 to n
5.          if A[i] + c > A[i] then
6.              c ← c + A[i];
7.          else
8.              c ← A[i];
9.          if c > m then
10.             m ← c;
11.     return m;
```

Какво връща той? Да се докаже формално.

Решение. Ще докажем, че $Kadane(A[1..n])$ връща най-голяма сума на непразен подмасив (последователни елементи) на $A[1..n]$.

Инвариант

При всяко k-то достигане на ред 4 имаме:

- $i = k + 1$
- c е най-голяма сума на непразен подмасив на $A[1..k]$, завършващ в индекс k
- m е най-голяма сума на непразен подмасив на $A[1..k]$

База. При $k = 1$-во достигане на ред 4 имаме:

- $i \overset{\text{ред } 4}{=} 2 = k + 1$
- c изпълнява инварианта
- m изпълнява инварианта

Поддръжка. Нека са изпълнени за някое k-то непоследно достигане на ред 4 следните:

- $i_{old} = k + 1$
- c_{old} е най-голяма сума на непразен подмасив на $A[1..k]$, завършващ в индекс k
- m_{old} е най-голяма сума на непразен подмасив на $A[1..k]$

Ще докажем за $(k + 1)$-во достигане на ред 4:

- $i_{new} = i_{old} + 1 \overset{\text{ИХ}}{=} (k + 1) + 1$

- Ще използваме следния факт наготово: $c_{new} = max\{c_{old} + A[i_{old}], A[i_{old}]\}$ (докажете). Допускаме, че c_{new} не е най-голяма сума на непразен подмасив на $A[1..k+1]$, завършваща в индекс $k+1$ и нека означим с t тази най-голяма сума. Тоест имаме, че $t > c_{new} = max\{c_{old} + A[i_{old}], A[i_{old}]\} \geq c_{old} + A[i_{old}]$. Като прехвърлим $A[i_{old}]$ от другата страна получаваме $t - A[i_{old}] > c_{old}$. Допуснахме, че t е най-голяма сума сума на непразен подмасив на $A[1..k+1]$, завършващ в индекс $k+1$. Тогава $t - A[i_{old}] \overset{\text{ИХ}}{=} t - A[k+1]$ е сума на **(потенциално празен)** подмасив на $A[1..k]$, завършващ в индекс k. Разглеждаме два случая:

  сл.1 (подмасива е празен)
  Тогава имаме, че t е сума само на последния елемент, т.е. $t = A[k+1]$. Знаем, че $c_{new} = max\{c_{old} + A[i_{old}], A[i_{old}]\}$. Нека разгледаме два подслучая:

  сл.1.1 ($c_{old} > 0$)
  Тогава имаме $c_{new} = \{c_{old} + A[i_{old}], A[i_{old}]\} = c_{old} + A[i_{old}] > A[i_{old}] \overset{\text{ИХ}}{=} A[k+1] = t$. Тоест излезе, че $c_{new} > t$ - противоречие (с това, че $t > c_{new}$).

  сл.1.2 ($c_{old} \leq 0$)
  Тогава имаме $c_{new} = \{c_{old} + A[i_{old}], A[i_{old}]\} = A[i_{old}] \overset{\text{ИХ}}{=} A[k+1] = t$. Тоест излезе, че $c_{new} = t$ - противоречие (с това, че $t > c_{new}$).

  сл.2 (подмасива не е празен)
  Тогава имаме, че $t - A[k+1]$ е сума на непразен подмасив на $A[1..k]$, завършваща в индекс k и знаем още, че $t - A[k+1] > c_{old}$. От друга страна от ИХ имаме, че c_{old} е максима сума на непразен подмасив на $A[1..k]$, завършваща в индекс k, т.е. $c_{old} \geq t - A[k+1] = t - A[i_{old}]$ - противоречие (с това, че $t - A[i_{old}] > c_{old}$).

  Следователно c_{new} най-голяма сума на непразен подмасив на $A[1..k+1]$, завършващ в индекс $k+1$.

- Ще използваме следния факт наготово: $m_{new} = max\{c_{new}, m_{old}\}$ (докажете). Опитваме се да докажем, че m_{new} е максимална сума на непразен подмасив на $A[1..k+1]$. Разглеждаме следните два случая:

  сл.1 ($A[k+1]$ участва в някоя максимална сума на непразен подмасив на $A[1..k+1]$)
  Тогава е ясно, че c_{new} е тази сума. За пълнота разглеждаме два подслучая:

  сл.1.1 (има точно една максимална сума на непразен подмасив на $A[1..k+1]$)
  Тогава ще влезем в тялото на if-а на ред 9-10 и ще имаме $m_{new} = c_{new}$.

  сл.1.2 (има повече от една)
  Тогава няма да влезем в тялото на същия if и ще имаме $m_{new} = m_{old} (= c_{new})$.

  сл.2 ($A[k+1]$ не участва в никоя максимална сума на непразен подмасив на $A[1..k+1]$)
  Тогава имаме $m_{new} = m_{old}$, но от ИХ за m_{old} директно имаме твърдението за m_{new}.

  Следователно m_{new} е най-голяма сума на непразен подмасив на $A[1..k+1]$

Терминация. При $k = n$ за първия път не влизаме в тялото на for-а. Директно връщаме m, което от инварианта знаем, че е максимална сума на непразен подмасив на $A[1..n]$.

Задача 2.6. Дадена е кутия с 53 сини топки и 42 червени топки. В кутията няма други топки. Извън кутията имаме неограничен брой сини и червени топки. Даден е следния алгоритъм:

```
1. AlgBall() :
2.     while не остане една топка в кутията do
3.         извади две случайни топки;
4.         if двете топки са еднакъв цвят then
5.             добави една червена;
6.         else
7.             добави една синя;
```

Какъв е цвета на топката, която е останала самичка?

Решение. Ще докажем, че цвета на топката, която е останала е син.

Инвариант

При всяко k-то достигане на ред 2 имаме:

- броя топки в кутията е $max\{95-k+1, 1\}$
- броя сини топки в кутията е нечетен

Докажете за упражнение инварианта и покажете как от него следва цвета на останалата самичка топка!

Задача 2.7. Даден е следния алгоритъм:

```
1. Pred(A[1..n]) : // n ∈ ℕ0 , A ∈ ℝ^n
2.     for i ← 1 to n − 1
3.         for j ← i + 1 to n
4.             if A[i] = A[j] then
5.                 return TRUE;
6.     return FALSE;
```

Какво връща той? Да се докаже формално.

Решение. Ще докажем, че $Pred(A[1..n])$ връща истина тогава и само тогава когато има повтарящи се елементи (и лъжа иначе) в $A[1..n]$. За целта ще ни трябват два инварианта - един за външния цикъл и един за вътрешния цикъл.

Инвариант: Вътрешен цикъл

При всяко k-то достигане на ред 3 имаме, че елементите $A[i+1], \ldots, A[i+k-1]$ са различни от $A[i]$ и $i < j$

Инвариант: Външен цикъл

При всяко k-то достигане на ред 2 имаме, че елементите $A[1], \ldots, A[k-1]$ са уникални в $A[1..n]$

Доказателството на двата инварианта остава за упражнение. Сега остана да използваме инвариантите за да докажем коректността. Имаме две възможни места за изход от програмата - ред 5 и ред 6. Нека да ги разгледаме:

сл.1 (излизаме през ред 5)
От вътрешния инвариант, знаем че $i < j$. Също така знаем, че $A[i] = A[j]$ тъй като сме в тялото на if от ред 4-5. Тоест имаме два еднакви елемента в масива и връщаме истина, к.т.д.д. (Тук **измамихме**! Можете ли да откриете проблема и да го оправите?)

сл.2 (излизаме през ред 5)
От външния инвариант директно следва, че в масива всички елементи са уникални и връщаме лъжа, к.т.д.д.

Забележка. При ограничено време на контролно приоритизирайте доказателството на най-външния инвариант.

Забележка 2.5: Инварианта не е всичко

Понякога трябва бонус работа (след инварианта) за да се докаже коректност на алгоритъм! Примери за това са **Задача 2.6** и **Задача 2.7**.

Глава 3

Анализ на сложността на алгоритми

3.1 Итеративни алгоритми

Изложените по-долу суми, може да се ползват без доказателство на контролни и домашни:

$$\sum_{i=1}^{n} 1 = n = \theta(n) \qquad \sum_{i=1}^{n} \theta(1) = \theta(n) \qquad \sum_{i=c}^{n} = n - c + 1 = \theta(n)$$

$$\sum_{i=c}^{n} \theta(1) = \theta(n) \qquad \sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \theta(n^2) \qquad \sum_{i=1}^{n} \theta(i) = \theta(n^2)$$

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} = \theta(n^3) \qquad \sum_{i=1}^{n} \theta(i^2) = \theta(n^3) \qquad \sum_{\substack{i=1 \\ i=i+k}}^{n} 1 = \lfloor \frac{n}{k} \rfloor = \theta(n)$$

$$\sum_{\substack{i=1 \\ i=i+k}}^{n} \theta(1) = \theta(n) \qquad \sum_{i=1}^{n} \frac{1}{i} = \theta(ln(n)) \qquad \sum_{i=1}^{n} \theta(\frac{1}{i}) = \theta(ln(n))$$

Забележка. Употребата на тета-нотацията вляво на $=$ се нарича ***анонимна функция***

3.1.1 Интегрален критерий

В настоящия курс ще разглеждаме само частен случай на интегралния критерий, който ще ни бъде достатъчен. Това е един от най-силните инструменти, с който ще разполагаме за да намирате асимптотика на дадена сума.

Теорема 3.1: Интегрален критерий (частен случай)

Нека f е асимптотично положителна функция и нека $n_0 \in \mathbb{N}_0$ е свидетел за това. Нека $\exists m > 0 \, \exists M > 0 \, \forall n > n_0 : mf(n) \leq \min\limits_{x \in [n,n+1]} f(x) \leq \max\limits_{x \in [n,n+1]} f(x) \leq Mf(n)$.

Тогава е изпълнено: $\sum\limits_{i=n_0}^{n} f(i) \asymp \int_{n_0}^{n} f(x)dx$.

Забележка. Приложен интегрален критерий без посочени конкретни m, M и n_0 и/или без обосновка на трите неравенства, носи **нула** точки!

3.1.1.1 Добре разпределени функции

Една функция ще наричаме ***добре разпределена*** (локална дефиниция), ако отговаря на допълнителното условие в **Теорема 3.1** за съществуване на такива константи m и M. Нека да разгледаме малко примери да придобием интуиция кои функции са ***добре разпределени***:

Пример 3.1. $f(x) = x^2$
Ще проверим, че условието

$$mn^2 \leq \min_{x\in[n,n+1]} f(x) \leq \max_{x\in[n,n+1]} f(x) \leq Mn^2 \tag{3.1}$$

е изпълнено за $m = 1, M = 4, \forall n \geq n_0 = 1$:

$$1n^2 \leq n^2 \leq (n+1)^2 \leq 4n^2$$

Първите две неравенства очевидно са верни. Остава да проверим:

$$(n+1)^2 \leq 4n^2$$

След като разкрием скобите получаваме:

$$n^2 + 2n + 1 \leq 4n^2$$

или още

$$3n^2 - 2n - 1 \geq 0$$

Сега намираме корените $x_1 = -\frac{1}{3}, x_2 = 1$ и виждаме, че $3n^2 - 2n - 1 \geq 0$ за $n \geq 1$, което искахме да докажем.

Разбира се, не е задължително да има единствени такива m, M, n_0. Даже напротив - ако има едно решение, то има безкрай много решения. Да разгледаме друго решение. Ще се убедим, че е изпълнено за $m = \frac{1}{3}, M = 1.005, \forall n \geq n_0 = 41$:

$$\frac{1}{3}n^2 \leq n^2 \leq (n+1)^2 \leq 1.005n^2$$

Отново първите две неравенства очевидно са верни. Остава да проверим:

$$(n+1)^2 \leq 1.005n^2$$

или още

$$0.05n^2 - 2n - 1 \geq 0$$

Сега намираме корените $x_1 = 20 - 2\sqrt{105} \approx -0.4939, x_2 = 20 + 2\sqrt{105} \approx 40.4939$ и виждаме, че $0.05n^2 - 2n - 1 \geq 0$ за $n \geq n_0 = 41 > x_2 \approx 40.4939$.

В случая кое да е $m \in (0, 1]$ и кое да е $M \in (1, +\infty)$ ни вършат работа, като в зависимост от избора ни на M, трябва да изберем подходящо n_0.

Пример 3.2. $f(x) = x^\alpha$
В зависимост от α ще проверим, че условието 3.1 е изпълнено за

сл.1 $(\alpha \geq 0)$ $m = 1, M = 2^\alpha, n_0 = 1$
Тоест трябва да се убедим, че следните неравенства са вярни:

$$1n^\alpha \leq n^\alpha \leq (n+1)^\alpha \leq 2^\alpha n^\alpha$$

Отново първите две са очевидни. Остава да се убедим само в последното неравенство:

$$(n+1)^\alpha \leq (2n)^\alpha$$

Тъй като $n_0 > 0$ и работим с $n \geq n_0$, то $2n > 0$ и можем да разделим на него (т.е. не делим на нула и не се сменя посоката на неравенството):

$$\left(\frac{n+1}{2n}\right)^\alpha \leq 1$$

Лесно съобразяваме, че при $n \geq n_0 = 1$ и $\alpha \geq 0$ неравенството е изпълнено:

$$\left(\underbrace{\frac{n+1}{2n}}_{\leq 1}\right)^\alpha \leq 1$$

сл.2 $(\alpha < 0)$ $m = 2^\alpha, M = 1, n_0 = 1$
Тоест трябва да се убедим, че следните неравенства са вярни:

$$2^\alpha n^\alpha \leq (n+1)^\alpha \leq n^\alpha \leq 1n^\alpha$$

Ще проверим първото неравенство, другите две са очевидни:

$$(2n)^\alpha \leq (n+1)^\alpha$$

Тъй като $n_0 > 0$ и работим с $n \geq n_0$, то $2n > 0$ и можем да разделим на него (т.е. не делим на нула и не се сменя посоката на неравенството):

$$\left(\frac{n+1}{2n}\right)^\alpha \geq 1$$

Лесно съобразяваме, че при $n \geq n_0 = 1$ и $\alpha < 0$ неравенството е изпълнено:

$$\left(\underbrace{\frac{n+1}{2n}}_{\leq 1}\right)^\alpha = \left(\underbrace{\frac{2n}{n+1}}_{\geq 1}\right)^{|\alpha|} \geq 1$$

Пример 3.3. $f(x) = \alpha^n,\ \alpha > 0$
В зависимост от α ще проверим, че условието 3.1 е изпълнено за

сл.1 $(\alpha \geq 1)$ $m = 1, M = \alpha, n_0 = 1$
Тоест трябва да се убедим, че следните неравенства са вярни:

$$1\alpha^n \leq \alpha^n \leq \alpha^{n+1} \leq \alpha\alpha^n$$

Очевидно и трите неравенства са верни.

сл.2 $(0 < \alpha < 1)$ $m = \alpha, M = 1, n_0 = 1$
Тоест трябва да се убедим, че следните неравенства са верни:

$$\alpha\alpha^n \leq \alpha^{n+1} \leq \alpha^n \leq 1\alpha^n$$

Очевидно и трите неравенства са верни.

Пример 3.4. $f(x) = x^x$
Това е пример за функция, която **не** е добре разпределена. Убедете се защо!

Пример 3.5. $f(x) = x!$
Това е друг пример за функция, която **не** е добре разпределена. Убедете се защо!

3.1.1.2 Приложения

Приложение 3.1. $f(x) = x^\alpha$
Очевидно функията е асимптотично положителна и в **Пример 3.2** доказахме, че тя е добре разпределена. Тогава може да приложим интегралния критерий и в зависимост какво n_0 сме избрали получаваме:

$$\sum_{i=n_0}^{n} i^\alpha \asymp \int_{n_0}^{n} x^\alpha dx = \begin{cases} \theta(n^{\alpha+1}) & , \alpha > -1 \\ \theta(ln(n)) & , \alpha = -1 \\ \theta(1) & , \alpha < -1 \end{cases}$$

Приложение 3.2. $f(x) = \alpha^x,\ \alpha > 0$
Очевидно функията е асимптотично положителна и в **Пример 3.3** доказахме, че тя е добре разпределена. Тогава може да приложим интегралния критерий и в зависимост какво n_0 сме избрали получаваме:

$$\sum_{i=n_0}^{n} \alpha^i \asymp \int_{n_0}^{n} \alpha^x \, dx$$

Сега разглеждаме 3 случая в зависимост от α:

сл.1 $(\alpha > 1)$

$$\int_{n_0}^{n} \alpha^x \, dx = \frac{\alpha^n - \alpha^{n_0}}{ln(\alpha)} = \theta(\alpha^n)$$

сл.2 $(\alpha = 1)$

$$\int_{n_0}^{n} \alpha \, dx = \alpha(n - n_0) = \theta(n)$$

сл.3 $(0 < \alpha < 1)$

$$\int_{n_0}^{n} \alpha^x \, dx = \frac{\alpha^n - \alpha^{n_0}}{ln(\alpha)} = \theta(1)$$

Тъй като числителя $\alpha^n - \alpha^{n_0}$ и знаменателя $ln(\alpha)$ са строго по-малки от 0, то цялата дроб е строго положителна. Тогава $\lim\limits_{n\to\infty} \frac{\alpha^n - \alpha^{n_0}}{ln(\alpha)} = \frac{-\alpha^{n_0}}{ln(\alpha)} > 0$ е просто константа, откъдето имаме асимптотика $\theta(1)$.

Тоест получихме:

$$\sum_{i=n_0}^{n} \alpha^i = \begin{cases} \theta(\alpha^n) & , \alpha > 1 \\ \theta(n) & , \alpha = 1 \\ \theta(1) & , 0 < \alpha < 1 \end{cases}$$

Приложение 3.3. Каква е асимптотиката на:

(а) $\sum\limits_{i=1}^{n} \frac{1}{i}$ (б) $\sum\limits_{i=1}^{n} \frac{1}{\sqrt{i}}$ (в) $\sum\limits_{i=1}^{n} \frac{1}{i^2}$

Решение. Това е частен случай на **Приложение 3.1**. Тоест имаме:

(а) $\sum\limits_{i=1}^{n} \frac{1}{i} = \theta(ln(n))$ (б) $\sum\limits_{i=1}^{n} \frac{1}{\sqrt{i}} = \theta(\sqrt{n})$ (в) $\sum\limits_{i=1}^{n} \frac{1}{i^2} = \theta(1)$

Приложение 3.4. Каква е асимптотиката на $\sum\limits_{i=1}^{n} ln(i)$?

Решение. Очевидно е, че функцията $f(x) = ln(x)$ е асимптотично положителна. Остана да покажем, че тя е добре разпределена. Тоест трябва да намерим такива $m, M > 0$ и $n_0 \in \mathbb{N}_0$, че да е вярно неравенство 3.1. Ще покажем, че $m = 1, M = 2, n_0 = 2$ са свидели:

$$1ln(n) \leq ln(n) \leq ln(n+1) \leq 2ln(n)$$

Очевидно първите две неравенства са изпълнени. Остава да проверим третото:

$$ln(n+1) \leq 2ln(n) = ln(n^2)$$

Нека се убедим първо, че следното неравенство е изпълнено за $n \geq n_0 = 2$:

$$n + 1 \leq n^2$$

или още

$$n^2 - n - 1 \geq 0$$

Сега намираме корените $x_1 = \frac{1-\sqrt{5}}{2} \approx -0.618, x_2 = \frac{1+\sqrt{5}}{2} \approx 1.618$ и виждаме, че $n^2 - n - 1 \geq 0$ за $n \geq n_0 = 2 > x_2 \approx 1.618$. Знаем, че логаритмуването на асимптотично положителни и неограничени отгоре функции запазва посоката на неравенството, откъдето получаваме:

$$ln(n+1) \leq ln(n^2) = 2ln(n)$$

Сега вече може да приложим интегралния критерий, откъдето получаваме:

$$\sum_{i=2}^{n} ln(i) \asymp \int_2^n ln(x)\,dx = ln(x)x\Big|_2^n - \int_2^n x\,d(ln(x)) = nln(n) - 2ln(2) - \int_2^n \frac{x}{x}\,dx =$$
$$= nln(n) - 2ln(2) - x\Big|_2^n = nln(n) - n + 2 - 2ln(2) = \theta(n.ln(n))$$

Забележете, че в оригиналната задача, сумата започваше от 1, а интегралния критерий ни даде асимптотика от 2 нагоре. Това обаче не е проблем, тъй като сме изпуснали $n_0 - 1$ на брой крайни събираеми (което е константа). Тоест имаме:

$$\sum_{i=1}^{n} ln(i) = const + \sum_{i=2}^{n} ln(i) = const + \theta(n.ln(n)) = \theta(n.ln(n))$$

В случая даже изпуснатото събираемо е $ln(1) = 0$, но в общия случай това не е така.

Приложение 3.5. Каква е асимптотиката на $\sum_{i=1}^{n} \frac{ln(i)}{i}$?

Решение. За упражнение докажете (използвайки интегралния критерий), че:

$$\sum_{i=1}^{n} \frac{ln(i)}{i} \asymp \theta(ln^2(n))$$

3.1.2 Задачи

Основната идея е да заместим всяка атомарна операция с константа (или за по-лесно единица - не се отразява на асимптотиката), а циклите с математическа сума. Нека разгледаме няколко примерни задачи:

Задача 3.1. Даден е следния алгоритъм:

```
1. Func(n) : // n ∈ ℕ⁺
2. |   print 'a';
3. |   for i ← 1 to n
4. |   |   print 'b';
5. |   fastprint 'c';
6. |   print 'd';
```

Каква е сложността му по време (спрямо големината на входа n)?

Решение. Сложността му по време е $c_{print} + \sum_{i=1}^{n}(c_{print} + c_{checkend} + c_{increment}) + c_{fastprint} + c_{print} = 2c_{print} + n(c_{print} + c_{checkend} + c_{increment}) + c_{fastprint} = \theta(n)$.

Забележка. За чистота се разбираме да пишем 1 вместо c_{name} за всички атомарни операции.

Задача 3.2. Даден е следния алгоритъм:

```
1. Func(n) : // n ∈ ℕ⁺
2. |  for i ← 1 to n
3. |  |  for j ← 1 to n
4. |  |  |  print 'a';
```

Каква е сложността му по време (спрямо големината на входа n)?

Решение. Сложността му по време е $\sum_{i=1}^{n}\sum_{j=1}^{n} 1 = \sum_{i=1}^{n} n = n\sum_{i=1}^{n} 1 = n.n = \theta(n^2)$.

Задача 3.3. Даден е следния алгоритъм:

```
1. Func(n) : // n ∈ ℕ⁺
2. |  for i ← 1 to n
3. |  |  for j ← 1 to i
4. |  |  |  print 'a';
```

Каква е сложността му по време (спрямо големината на входа n)?

Решение. Сложността му по време е $\sum_{i=1}^{n}\sum_{j=1}^{i} 1 = \sum_{i=1}^{n} i = \frac{(n+1)n}{2} = \theta(n^2)$.

Задача 3.4. Даден е следния алгоритъм:

```
1. Func(n) : // n ∈ ℕ⁺
2. |  for i ← 1 to n
3. |  |  for j ← 1 to n with step i
4. |  |  |  print 'a';
```

Каква е сложността му по време (спрямо големината на входа n)?

Решение. Сложността му по време е $\sum_{i=1}^{n}\sum_{\substack{j=1 \\ j=j+i}}^{n} 1 = \sum_{i=1}^{n}\lfloor\frac{n}{i}\rfloor \asymp \sum_{i=1}^{n}\frac{n}{i} = n\sum_{i=1}^{n}\frac{1}{i} \asymp nln(n) = \theta(n.ln(n))$.

Задача 3.5. Даден е следния алгоритъм:

```
1. Func(n) : // n ∈ ℕ⁺
2.     for i ← 1 to n
3.         for j ← i to 2^i
4.             if j < n then
5.                 for k ← 1 to n with step i
6.                     print 'a';
7.             print 'b';
```

Каква е сложността му по време (спрямо големината на входа n)?

Решение. Сложността му по време е:

$$\underbrace{\sum_{i=1}^{n}\sum_{j=i}^{2^i} 1}_{\text{print 'b'}} + \underbrace{\sum_{i=1}^{\lfloor log(n)\rfloor}\sum_{j=i}^{2^i}\sum_{\substack{k=1\\k=k+i}}^{n} 1}_{\text{print 'a' } (i\leq\lfloor log(n)\rfloor)} + \underbrace{\sum_{i=\lfloor log(n)\rfloor+1}^{n}\sum_{j=i}^{n}\sum_{\substack{k=1\\k=k+i}}^{n} 1}_{\text{print 'a' } (i>\lfloor log(n)\rfloor)} \tag{3.2}$$

Нека да разгледаме трите суми поотделно. Да започнем с първата:

$$\sum_{i=1}^{n}\sum_{j=i}^{2^i} 1 = \sum_{i=1}^{n}\Big(2^i - i + 1\Big) = \sum_{i=1}^{n} 2^i - \sum_{i=1}^{n} i + \sum_{i=1}^{n} 1 \asymp 2^n - n^2 + n = \theta(2^n)$$

Сега да разгледаме втората (като премахнем закръглянето - не променя асимптотиката):

$$\begin{aligned}\sum_{i=1}^{log(n)}\sum_{j=i}^{2^i}\sum_{\substack{k=1\\k=k+i}}^{n} 1 &\asymp \sum_{i=1}^{log(n)}\sum_{j=i}^{2^i}\frac{n}{i} = n\sum_{i=1}^{log(n)}\left(\frac{1}{i}\sum_{j=i}^{2^i} 1\right) = n\sum_{i=1}^{log(n)}\left(\frac{1}{i}\Big(2^i - i + 1\Big)\right) = \\ &= n\left(\sum_{i=1}^{log(n)}\frac{2^i}{i} - \sum_{i=1}^{log(n)}\frac{i}{i} + \sum_{i=1}^{log(n)}\frac{1}{i}\right) \leq n\left(\sum_{i=1}^{log(n)} 2^i - \sum_{i=1}^{log(n)} 1 + \sum_{i=1}^{log(n)}\frac{1}{i}\right) \asymp \\ &\asymp n\Big(2^{log(n)} - log(n) + ln(log(n))\Big) \asymp n\Big(n - log(n) + log^{(2)}(n)\Big) \asymp n^2\end{aligned}$$

В крайна сметка за втората сума получихме:

$$\sum_{i=1}^{log(n)}\sum_{j=i}^{2^i}\sum_{\substack{k=1\\k=k+i}}^{n} 1 = O(n^2)$$

Обърнете внимание, че е $O(n^2)$, а не $\theta(n^2)$ - това е заради употребата на $\leq$. Разбира се може в действителност да е $\theta(n^2)$, но $O(n^2)$ ни е достатъчно (след малко ще видим защо) да определим сложността по време на 3.2. Остана да разгледаме третата сума (отново махаме закръглянето и константата $+1$ - не променят асимптотиката):

$$\begin{aligned}\sum_{i=log(n)}^{n}\sum_{j=i}^{n}\sum_{\substack{k=1\\k=k+i}}^{n} 1 &\asymp \sum_{i=log(n)}^{n}\sum_{j=i}^{n}\frac{n}{i} = n\sum_{i=log(n)}^{n}\frac{1}{i}\sum_{j=i}^{n} 1 = n\sum_{i=log(n)}^{n}\left(\frac{1}{i}\Big(n - i + 1\Big)\right) = \\ &= n\left(n\sum_{i=log(n)}^{n}\frac{1}{i} - \sum_{i=log(n)}^{n}\frac{log(n)}{i} + \sum_{i=log(n)}^{n}\frac{1}{i}\right) = \cdots = \theta(n^2 log(n))\end{aligned}$$

Тоест сложността по време на функцията е $\theta(2^n) + O(n^2) + \theta(n^2 log(n)) = \theta(2^n)$.

3.2 Рекурсивни алгоритми

Сложността на рекурсивни алгоритми ще определяме посредством рекурентни уравнения. По тази причина рекурентните уравнения, които ще разглеждаме ще са със строго намаляващ аргумент, всички събираеми отдясно на $=$ ще се срещат с положителен знак и всички основи на експоненти в нехомогенната част ще бъдат положителни. Също така да обърнем внимание, че за всяко фиксирано $n_0 \in \mathbb{N}_0$ е изпълнено: $\forall n \leq n_0\,(T(n) = \theta(1))$, където $T(n)$ е рекурентно уравнение. Може да си мислим за това n_0 като базата на рекурсивния алгоритъм. За удобство ще нагласяме n_0 да е някое малко число - например $0, 1, 2$ - в зависимост от конкретната задача. Съставянето на рекурентно уравнение за сложността на рекурсивен алгоритъм е тривиално в повечето случаи. Поради тази причина ще се съсредоточим върху методи за намиране на асимптотиката на рекурентни уравнения.

3.2.1 Характеристично уравнение

Започваме с метод, който ви е добре познат от ДСТР. За разлика от ДСТР, тук не ни интересува точното решение, а само асимптотиката.

Задача 3.6. Каква е асимптотиката на $T(n) = 4T(n-2) + n2^n + 4.3^n$?

Решение.

- (Хомогенна част)
  $x^2 = 4 \;\mapsto\; x_{1,2} = \pm 2 \;\mapsto\; \{2, -2\}_M$
- (Нехомогенна част)
  - $n2^n \;\mapsto\; \{2, 2\}_M$
  - $4.3^n \;\mapsto\; \{3\}_M$

Като обединим всички мултимножества получаваме: $\{-2, 2, 2, 2, 3\}_M$. Оттук получаваме, че $T(n) = c_1(-2)^n + c_2 2^n + c_3 n 2^n + c_4 n^2 2^n + c_5 3^n = \theta(3^n)$.

Забележка. Забележете, че поради естеството на рекурентните уравнения (строго намаляващ аргумент, положителен знак отдясно на $=$ и положителна основа на експонентата в нехомогенната част), то ни е гарантирано, че най-голямото (по модул) число в крайното мултимножество ще е положително и при това ще се среща със строго положителна константа в явния запис на рекурентното уравнение. Поради тази причина в **Задача 3.6** сме сигурни, че $c_5 > 0$ (и че $\max\{|-2|, |2|, |2|, |2|, |3|\} = 3 \in \{-2, 2, 2, 2, 3\}$).

Задача 3.7. Каква е асимптотиката на $T(n) = 2T(n-1) + T(n-2)$?

Решение. Тук имаме само хомогенна част: $x^2 = 2x+1 \;\mapsto\; x_{1,2} = 1 \pm \sqrt{2} \;\mapsto\; \{1+\sqrt{2}, 1-\sqrt{2}\}_M$. Оттук получаваме $T(n) = c_1(1+\sqrt{2})^n + c_2(1-\sqrt{2})^n = \theta((1+\sqrt{2})^n)$.

Задача 3.8. Каква е асимптотиката на $T(n) = T(n-2) + T(n-4) + \cdots + \underbrace{T(n\%2)}_{T(0) \text{ или } T(1)}$?

Решение. Това характеристично уравнение се решава със следния трик: $T(n) - T(n-2) = \underbrace{T(n-2) + T(n-4) + \cdots + T(n\%2)}_{T(n)} - \underbrace{(T(n-4) + T(n-6) + \cdots + T(n\%2))}_{T(n-2)} = T(n-2)$. Тоест получихме, че $T(n) - T(n-2) = T(n-2)$ или още $T(n) = 2T(n-2)$. Решаваме характеристичното уравнение $x^2 = 2$ и получаваме $x_{1,2} = \pm\sqrt{2} \mapsto \{\sqrt{2}, -\sqrt{2}\}_M$. Оттук имаме $T(n) = c_1(\sqrt{2})^n + c_2(-\sqrt{2})^n = \theta((\sqrt{2})^n)$.

3.2.2 Развиване

Метода за намиране асимптотика на рекурентно уравнение чрез развиване е най-мощния, който ще разгледаме в текущия курс, но и най-хамалския. Изисква сравнително много писане и известна доза *умен съм бил сетил съм се* (зависи от задачата). Самия метод не е формален. Формализацията изисква доказателство чрез индукция, което е трудната част. Идеята е много проста - развиваме докажо не заподозреем сложността и след това доказваме формално.

Задача 3.9. Каква е асимптотиката на $T(n) = T(n-1) + n$?

Решение. Тази задача може да се реши чрез характеристично уравнение, но сега ще го докажем чрез развиване и индукция.

$$\begin{aligned} T(n) &= T(n-1) + n = T(n-2) + (n-1) + n = \\ &= T(n-3) + (n-2) + (n-1) + n = \cdots = \\ &= T(0) + 1 + 2 + \cdots + (n-1) + n = T(0) + \theta(n^2) = \theta(n^2) \end{aligned}$$

Забележка

Дотук **НЕ** сме доказали нищо! Само сме заподозрели отговора! Проблема е в това, че употребата на "..." е изцяло неформална.. може да сме сбъркали нещо на ум!

Нека $n_{st} \in \mathbb{N}_0$ е достатъчно голямо[1].

- $T(n) = O(n^2)$

  По дефиниция имаме $O(n^2) = \{f \mid \exists c > 0\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0$ е изпълнено $0 \leq f(n) \leq c.n^2\}$. Тоест търсим константи $c > 0$ и $n_0 \in \mathbb{N}_0$ такива, че $\forall n \geq n_0\, (0 \leq T(n) \leq c.n^2)$. Както се разбрахме в началото на 3.2, сложността на рекурсивните алгоритмите, които ще разглеждаме в текущия курс, изпълняват условието $T(n) \geq 0$. Тоест търсим константи $c > 0$ и $n_0 \in \mathbb{N}_0$ такива, че $\forall n \geq n_0\, (T(n) \leq c.n^2)$.

  Нека $b = \max\left\{\frac{T(1)}{1^2}, \frac{T(2)}{2^2}, \frac{T(3)}{3^2}, \ldots, \frac{T(n_{\text{st}}-1)}{(n_{\text{st}}-1)^2}, 1\right\}$[2]. (виж **Забележка 3.1**)

  Ще докажем с индукция по n, че за $c = b$ и някое n_0 (което ще установим по-надолу) е изпълнено $\forall n \geq n_0\, (T(n) \leq c.n^2)$.

[1] Надолу в доказателството ще въведем n_0 и ще изискваме $n_0 < n_{st}$.

[2] Надолу ще установим, защо добавяме тази единица.

База. Нека $k \in \{1, 2, \ldots, n_{st} - 1\}$. Ще докажем, че $T(k) \leq b.k^2$.

От дефиницията на $b = \max\left\{\frac{T(1)}{1^2}, \frac{T(2)}{2^2}, \frac{T(3)}{3^2}, \ldots, \frac{T(n_{\text{st}}-1)}{(n_{\text{st}}-1)^2}, 1\right\}$ знаем, че $b \geq \frac{T(k)}{k^2}$ откъдето $b.k^2 \geq T(k)$ или още $T(k) \leq b.k^2$.

Индуктивна хипотеза. Нека допуснем, че е изпълнено $\forall m < n\,(T(m) \leq b.m^2)$.

Индуктивна стъпка. Ще докажем, че е изпълнено за n, тоест че $T(n) \leq b.n^2$.

$$T(n) \overset{\text{def}}{=} T(n-1) + n \overset{\text{ИХ}}{\leq} b(n-1)^2 + n = bn^2 - 2bn + b + n \overset{?}{\leq} bn^2$$

$$bn^2 - 2bn + b + n \overset{?}{\leq} bn^2$$

$$n(1 - 2b) + b \overset{?}{\leq} 0$$

Може да забележим, че при $1 - 2b \leq -b$ горното неравенство ще е изпълнено за всяко $n \geq 1$. Оттук си избираме $n_0 = 1$.

Остана да видим кога е изпълнено $1 - 2b \leq -b$. Очевидно е изпълнено когато $b \geq 1$. Това сме си го подсигурили от дефиницията на $b = \max\{\ldots, 1\} \Rightarrow b \geq 1$.

Тоест доказахме, че за $c = b \wedge n_0 = 1$ е изпълнено $\forall n \geq n_0\,(0 \leq T(n) \leq c.n^2)$. Казано с други думи $T(n) = O(n^2)$.

- $T(n) = \Omega(n^2)$

  По дефиниция имаме $\Omega(n^2) = \{f \mid \exists c > 0\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0$ е изпълнено $0 \leq c.n^2 \leq f(n)\}$. Тоест търсим константи $c > 0$ и $n_0 \in \mathbb{N}_0$ такива, че $\forall n \geq n_0\,(c.n^2 \leq T(n))$.

  Нека $b = \min\left\{\frac{T(1)}{1^2}, \frac{T(2)}{2^2}, \frac{T(3)}{3^2}, \ldots, \frac{T(n_{\text{st}}-1)}{(n_{\text{st}}-1)^2}, \frac{1}{2}\right\}$. (виж **Забележка 3.1**)

  Ще докажем с индукция по n, че за $c = b$ и някое n_0 (което ще установим по-надолу) е изпълнено $\forall n \geq n_0\,(c.n^2 \leq T(n))$.

  База. Нека $k \in \{1, 2, \ldots, n_{st} - 1\}$. Ще докажем, че $b.k^2 \leq T(k)$.

  От дефиницията на $b = \min\left\{\frac{T(1)}{1^2}, \frac{T(2)}{2^2}, \frac{T(3)}{3^2}, \ldots, \frac{T(n_{\text{st}}-1)}{(n_{\text{st}}-1)^2}, \frac{1}{2}\right\}$ знаем, че $b \leq \frac{T(k)}{k^2}$ откъдето $b.k^2 \leq T(k)$.

  Индуктивна хипотеза. Нека допуснем, че е изпълнено $\forall m < n\,(b.m^2 \leq T(m))$.

  Индуктивна стъпка. Ще докажем, че е изпълнено за n, тоест че $b.n^2 \leq T(n)$.

$$T(n) \overset{\text{def}}{=} T(n-1) + n \overset{\text{ИХ}}{\geq} b(n-1)^2 + n = bn^2 - 2bn + b + n \overset{?}{\geq} bn^2$$

$$n(1 - 2b) + b \overset{?}{\geq} 0$$

  Може да забележим, че при $1 - 2b \geq 0$ горното неравенство ще е изпълнено за всяко $n \geq 0$. Оттук си избираме $n_0 = 1$ (ако изберем $n_0 = 0$ ще имаме $\frac{T(0)}{0}$).

  Остана да видим кога е изпълнено $1 - 2b \geq 0$. Очевидно е изпълнено когато $b \leq \frac{1}{2}$. Това сме си го подсигурили от дефиницията на $b = \min\{\ldots, \frac{1}{2}\} \Rightarrow b \leq \frac{1}{2}$.

  Тоест доказахме, че за $c = b \wedge n_0 = 1$ е изпълнено $\forall n \geq n_0\,(0 \leq c.n^2 \leq T(n))$. Казано с други думи $T(n) = \Omega(n^2)$.

Забележка 3.1: Константата c при доказване с индукция

Когато доказваме, че асимптотиката на рекурентно уравнение е $\varphi(n)$ чрез индукция, то константата винаги е от вида max (за O) или min (за Ω) на $\{\frac{T(n_0)}{\varphi(n_0)}, \frac{T(n_0+1)}{\varphi(n_0+1)}, \ldots, \frac{T(n_{st}-1)}{\varphi(n_{st}-1)}\}$, като внимаваме за избора на n_0 - да не разделим на нула (примерно ако $\varphi(n) = log(n)$, то тогава $\frac{T(1)}{\varphi(1)} = \frac{T(1)}{0}$). Потенциално може да трябва още един елемент към множеството. Нека разгледаме няколко примерни константи:

- $T(n) = O(n^2) \mapsto \max\left\{\frac{T(1)}{1^2}, \frac{T(2)}{2^2}, \frac{T(3)}{3^2}, \ldots, \frac{T(n_{\text{st}}-1)}{(n_{\text{st}}-1)^2}, ?\right\}$
- $T(n) = \Omega(log(n)) \mapsto \min\left\{\frac{T(2)}{log(2)}, \frac{T(3)}{log(3)}, \ldots, \frac{T(n_{\text{st}}-1)}{log(n_{st}-1)}, ?\right\}$
- $T(n) = O(2^n) \mapsto \max\left\{\frac{T(0)}{2^0}, \frac{T(1)}{2^1}, \frac{T(2)}{2^2}, \ldots, \frac{T(n_{\text{st}}-1)}{2^{(n_{st}-1)}}, ?\right\}$
- $T(n) = \Omega(n^3-4n^2+3n) \mapsto \min\left\{\frac{T(4)}{4^3-4.4^2+3.4}, \frac{T(5)}{5^3-4.5^2+3.5}, \ldots, \frac{T(n_{\text{st}}-1)}{(n_{st}-1)^3-4(n_{st}-1)^2+3(n_{st}-1)}, ?\right\}$

Важно е да обърнем внимание, че при образуването на базата има разлика между $\Omega(n^3 - 4n^2 + 3n)$, $\Omega(n^3)$ и $\Omega(5n^3)$. Пример защо това е важно в O-посоката на **Задача 3.13**.

Задача 3.10. Каква е асимптотиката на $T(n) = T(n-1) + \frac{1}{n}$?

Решение. Нека да развием, докато не заподозреем асимптотиката.

$$\begin{aligned} T(n) &= T(n-1) + \frac{1}{n} = T(n-2) + \frac{1}{n-1} + \frac{1}{n} = \\ &= T(n-3) + \frac{1}{n-2} + \frac{1}{n-1} + \frac{1}{n} = \cdots = \\ &= T(0) + \frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n-1} + \frac{1}{n} = T(0) + \theta(ln(n)) = \theta(log(n)) \end{aligned}$$

Нека $n_{st} \in \mathbb{N}_0$ е достатъчно голямо.

- $T(n) = O(log(n))$

  По дефиниция имаме $O(log(n)) = \{f \mid \exists c > 0\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0\, (0 \leq f(n) \leq c.log(n))\}$. Тоест търсим $c > 0$ и $n_0 \in \mathbb{N}_0$ такива, че $\forall n \geq n_0\, (T(n) \leq c.log(n))$.

  Нека $b = \max\left\{\frac{T(2)}{log(2)}, \frac{T(3)}{log(3)}, \frac{T(4)}{log(4)}, \ldots, \frac{T(n_{\text{st}}-1)}{log(n_{\text{st}}-1)}, 1\right\}$.

  Ще докажем с индукция по n, че за $c = b$ и някое n_0 (което ще установим по-надолу) е изпълнено $\forall n \geq n_0\, (T(n) \leq c.log(n))$.

  База. Нека $k \in \{2, 3, \ldots, n_{st} - 1\}$. Ще докажем, че $T(k) \leq b.log(k)$.

  От дефиницията на $b = \max\left\{\frac{T(2)}{log(2)}, \frac{T(3)}{log(3)}, \frac{T(4)}{log(4)}, \ldots, \frac{T(n_{\text{st}}-1)}{log(n_{\text{st}}-1)}, 1\right\}$ знаем, че $b \geq \frac{T(k)}{log(k)}$ откъдето $b.log(k) \geq T(k)$ или още $T(k) \leq b.log(k)$.

  Индуктивна хипотеза. Нека допуснем, че е изпълнено $\forall m < n\, (T(m) \leq b.log(m))$.

Индуктивна стъпка. Ще докажем, че е изпълнено за n, тоест че $T(n) \leq b.log(n)$.

$$T(n) \overset{\text{def}}{=} T(n-1) + \frac{1}{n} \overset{\text{их}}{\leq} b.log(n-1) + \frac{1}{n} \overset{?}{\leq} b.log(n)$$

$$b.log(n-1) + \frac{1}{n} \overset{?}{\leq} b.log(n)$$

Тъй като $n \geq n_0 \in \mathbb{N}_0$, то може да умножим по n без да се обърне знака на неравенството.

$$b.n.log(n-1) + 1 \overset{?}{\leq} b.n.log(n)$$

$$1 \overset{?}{\leq} b.n.log(n) - b.n.log(n-1)$$

$$1 \overset{?}{\leq} b.n.log\left(\frac{n}{n-1}\right)$$

Антилогаритмуваме двете страни (със основа 2)

$$2 \overset{?}{\leq} \left(\frac{n}{n-1}\right)^{b.n}$$

От дефиницията на $b = \max\{\ldots, 1\} \Rightarrow b \geq 1$. Ясно е, че $\forall n \geq 1 \left(\frac{n}{n-1} \geq 1\right)$. Тоест имаме:

$$2 \overset{?}{\leq} \left(\frac{n}{n-1}\right)^{n} \leq \left(\frac{n}{n-1}\right)^{b.n}$$

$$2 \overset{?}{\leq} \left(\frac{n}{n-1}\right)^{n-1} \leq \left(\frac{n}{n-1}\right)^{n}$$

От анализа знаем, че $\lim\limits_{n\to\infty} \left(\frac{n}{n-1}\right)^{n-1} = e$ и че функцията $\left(\frac{n}{n-1}\right)^{n-1}$ е строго растяща. Тогава директно проверяваме, че $n_0 = 2$ ни върши работа

$$\forall n \geq n_0 \left(2 \leq \left(\frac{n}{n-1}\right)^{n-1} \leq \left(\frac{n}{n-1}\right)^{b.n}\right)$$

или още

$$\forall n \geq n_0 \left(2 \leq \left(\frac{n}{n-1}\right)^{b.n}\right)$$

Тъй като логаритмуването е монотонно растяща функция, то можем да запазим неравенството след логаритмуване на двете страни

$$\forall n \geq n_0 \left(1 \leq b.n.log\left(\frac{n}{n-1}\right)\right)$$

което трябваше да докажем.

- $T(n) = \Omega(log(n))$

  По дефиниция имаме $\Omega(log(n)) = \{f \mid \exists c > 0\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0\, (0 \leq c.log(n)) \leq f(n)\}$. Тоест търсим $c > 0$ и $n_0 \in \mathbb{N}_0$ такива, че $\forall n \geq n_0\, (c.log(n) \leq T(n))$.

  Нека $b = \min\left\{\frac{T(2)}{log(2)}, \frac{T(3)}{log(3)}, \frac{T(4)}{log(4)}, \ldots, \frac{T(n_{\text{st}}-1)}{log(n_{\text{st}}-1)}, \frac{1}{2}\right\}$.

  Ще докажем с индукция по n, че за $c = b$ и някое n_0 (което ще установим по-надолу) е изпълнено $\forall n \geq n_0\, (c.log(n) \leq T(n))$.

База. Нека $k \in \{2, 3, \ldots, n_{st} - 1\}$. Ще докажем, че $b.log(k) \leq T(k)$.

От дефиницията на $b = \min\left\{\frac{T(2)}{log(2)}, \frac{T(3)}{log(3)}, \frac{T(4)}{log(4)}, \ldots, \frac{T(n_{\text{st}}-1)}{log(n_{\text{st}}-1)}, \frac{1}{2}\right\}$ знаем, че $b \leq \frac{T(k)}{log(k)}$ откъдето $b.log(k) \leq T(k)$.

Индуктивна хипотеза. Нека допуснем, че е изпълнено $\forall m < n\,(b.log(m) \leq T(m))$.

Индуктивна стъпка. Ще докажем, че е изпълнено за n, тоест че $b.log(n) \leq T(n)$.

$$T(n) \overset{\text{def}}{=} T(n-1) + \frac{1}{n} \overset{\text{их}}{\geq} b.log(n-1) + \frac{1}{n} \overset{?}{\geq} b.log(n)$$

Аналогично на O-посоката стигаме до

$$2 \overset{?}{\geq} \left(\frac{n}{n-1}\right)^{b.n}$$

От дефиницията на $b = \min\{\ldots, \frac{1}{2}\} \Rightarrow b \leq \frac{1}{2}$. Ясно е, че $\forall n \geq 1\,\left(\frac{n}{n-1} \geq 1\right)$. Тоест имаме:

$$2 \overset{?}{\geq} \left(\frac{n}{n-1}\right)^{n/2} \geq \left(\frac{n}{n-1}\right)^{b.n}$$

Вдигаме на квадрат двете страни

$$4 \overset{?}{\geq} \left(\frac{n}{n-1}\right)^{n} = \left(\frac{n}{n-1}\right)\left(\frac{n}{n-1}\right)^{n-1}$$

От анализа имаме

$$4 \overset{?}{\geq} \left(\frac{n}{n-1}\right)e \geq \left(\frac{n}{n-1}\right)\left(\frac{n}{n-1}\right)^{n-1}$$

$$4(n-1) \overset{?}{\geq} ne$$

$$n(4-e) \overset{?}{\geq} 4$$

Директно проверяваме, че $n_0 = 4$ ни върши работа

$$\forall n \geq n_0\,(n(4-e) \geq 4)$$

$$\forall n \geq n_0 \left(4 \geq \left(\frac{n}{n-1}\right)e \geq \left(\frac{n}{n-1}\right)^{n}\right)$$

Тъй като коренуването е монотонно растяща функция, то можем да запазим неравенството след коренуване на двете страни

$$\forall n \geq n_0 \left(2 \geq \left(\frac{n}{n-1}\right)^{n/2} \geq \left(\frac{n}{n-1}\right)^{b.n}\right)$$

Тъй като логаритмуването е монотонно растяща функция, то можем да запазим неравенството след логаритмуване на двете страни

$$\forall n \geq n_0 \left(1 \geq b.n.log\left(\frac{n}{n-1}\right)\right)$$

което трябваше да докажем.

Задача 3.11. Каква е асимптотиката на $T(n) = 2T(n-1) + \frac{1}{n}$?

Решение. Нека да развием, докато не заподозреем асимптотиката.

$$\begin{aligned} T(n) &= 2T(n-1) + \frac{1}{n} = 4T(n-2) + \frac{2}{n-1} + \frac{1}{n} = \\ &= 8T(n-3) + \frac{4}{n-2} + \frac{2}{n-1} + \frac{1}{n} = \cdots = \\ &= 2^n T(0) + \frac{2^{n-1}}{1} + \frac{2^{n-2}}{2} + \cdots + \frac{2}{n-1} + \frac{1}{n} = \\ &= 2^n\theta(1) + \underbrace{2^n \sum_{i=1}^{n} \frac{1}{i2^i}}_{O(2^n)} = \theta(2^n) \end{aligned}$$

Докажете го формално чрез индукция.

Задача 3.12. Каква е асимптотиката на $T(n) = \frac{n}{n+1}T(n-1) + 1$?

Решение. Нека да развием, докато не заподозреем асимптотиката.

$$\begin{aligned} T(n) &= \frac{n}{n+1}T(n-1) + 1 = \frac{n-1}{n+1}T(n-2) + \frac{n}{n+1} + \frac{n+1}{n+1} = \\ &= \frac{n-2}{n+1}T(n-3) + \frac{n-1}{n+1} + \frac{n}{n+1} + \frac{n+1}{n+1} = \cdots = \\ &= \frac{1}{n+1}T(0) + \frac{2}{n+1} + \frac{3}{n+1} + \cdots + \frac{n}{n+1} + \frac{n+1}{n+1} = \\ &= \theta(n^{-1}) + \frac{1}{n+1}\underbrace{(2 + 3 + \cdots + (n+1))}_{\theta(n^2)} = \theta(n) \end{aligned}$$

Докажете го формално чрез индукция.

3.2.2.1 Засилване на индуктивната хипотеза

В някои ситуации доказването на индуктивната стъпка би било невъзможно без така наречното ***засилване на индуктивната хипотеза***. Какво представляво то се разбира най-добре с пример.

Задача 3.13. Каква е асимптотиката на $T(n) = 2T(n-1) + T(log(n)) + n$?

Решение. Заподозряваме, че асимптотиката е $T(n) = \theta(2^n)$.
Нека $n_{st} \in \mathbb{N}_0$ е достатъчно голямо.

- $T(n) = O(2^n)$

  По дефиниция имаме $O(2^n) = \{f \mid \exists c > 0\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0\, (0 \leq f(n) \leq c2^n)\}$. Тоест търсим константи $c > 0$ и $n_0 \in \mathbb{N}_0$ такива, че $\forall n \geq n_0\, (T(n) \leq c2^n)$.

Нека $b = \max\left\{\frac{T(0)}{2^0}, \frac{T(1)}{2^1}, \frac{T(2)}{2^2}, \ldots, \frac{T(n_{\text{st}}-1)}{2^{n_{\text{st}}-1}}\right\}$.

Ще докажем с индукция по n, че за $c = b$ и някое n_0 (което ще установим по-надолу) е изпълнено $\forall n \geq n_0\,(T(n) \leq c2^n)$.

База. Нека $k \in \{0, 1, \ldots, n_{st} - 1\}$. Ще докажем, че $T(k) \leq b2^k$.

От дефиницията на $b = \max\left\{\frac{T(0)}{2^0}, \frac{T(1)}{2^1}, \frac{T(2)}{2^2}, \ldots, \frac{T(n_{\text{st}}-1)}{2^{n_{\text{st}}-1}}\right\}$ знаем, че $b \geq \frac{T(k)}{2^k}$ откъдето $b2^k \geq T(k)$ или още $T(k) \leq b2^k$.

Индуктивна хипотеза. Нека допуснем, че е изпълнено $\forall m < n\,(T(m) \leq b2^m)$.

Индуктивна стъпка. Ще докажем, че е изпълнено за n, тоест че $T(n) \leq b2^n$.

$$T(n) \overset{\text{def}}{=} 2T(n-1) + T(log(n)) + n \overset{\text{ИХ}}{\leq} 2b2^{n-1} + b2^{log(n)} + n = b2^n + bn + n \overset{?}{\leq} b2^n$$

$$b2^n + bn + n \overset{?}{\leq} b2^n$$

$$bn + n \overset{?}{\leq} 0$$

Очевидно не е изпълнено за кое да е $b > 0$ и $n > 0$. Не стана.. ще трябва да засилим индуктивната хипотеза.

- $T(n) = O(2^n - \alpha^n)$, $\alpha \in (1, 2)$

  По дефиниция имаме $O(2^n - \alpha^n) = \{f \mid \exists c > 0\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0\,(0 \leq f(n) \leq c(2^n - \alpha^n))\}$. Тоест търсим константи $c > 0$ и $n_0 \in \mathbb{N}_0$ такива, че $\forall n \geq n_0\,(T(n) \leq c(2^n - \alpha^n))$.

  Нека $b = \max\left\{\frac{T(1)}{2^1-\alpha^1}, \frac{T(2)}{2^2-\alpha^2}, \ldots, \frac{T(n_{\text{st}}-1)}{2^{n_{\text{st}}-1}-\alpha^{n_{\text{st}}-1}}, 1\right\}$. (при нула знаменателя е $2^0 - \alpha^0 = 0$)

  Ще докажем с индукция по n, че за $c = b$ и някое n_0 (което ще установим по-надолу) е изпълнено $\forall n \geq n_0\,(T(n) \leq c(2^n - \alpha^n))$.

  База. Нека $k \in \{1, 2, \ldots, n_{st} - 1\}$. Ще докажем, че $T(k) \leq b(2^k - \alpha^k)$.

  От дефиницията на $b = \max\left\{\frac{T(1)}{2^1-\alpha^1}, \frac{T(2)}{2^2-\alpha^2}, \ldots, \frac{T(n_{\text{st}}-1)}{2^{n_{\text{st}}-1}-\alpha^{n_{\text{st}}-1}}, 1\right\}$ знаем, че $b \geq \frac{T(k)}{2^k-\alpha^k}$ откъдето $b(2^k - \alpha^k) \geq T(k)$ или още $T(k) \leq b(2^k - \alpha^k)$.

  Индуктивна хипотеза. Нека допуснем, че е изпълнено $\forall m < n\,(T(m) \leq b(2^m - \alpha^m))$.

  Индуктивна стъпка. Ще докажем, че е изпълнено за n, тоест че $T(n) \leq b(2^n - \alpha^n)$.

$$T(n) \overset{\text{def}}{=} 2T(n-1) + T(log(n)) + n \overset{\text{ИХ}}{\leq} 2b(2^{n-1} - \alpha^{n-1}) + b(2^{log(n)} - \alpha^{log(n)}) + n \overset{?}{\leq} b(2^n - \alpha^n)$$

$$2b(2^{n-1} - \alpha^{n-1}) + b(2^{log(n)} - \alpha^{log(n)}) + n \overset{?}{\leq} b(2^n - \alpha^n)$$

$$b2^n - 2b\alpha^{n-1} + bn - bn^{log(\alpha)} + n \overset{?}{\leq} b2^n - b\alpha^n$$

$$b(\alpha - 2)\alpha^{n-1} + bn - bn^{log(\alpha)} + n \overset{?}{\leq} 0 \tag{3.3}$$

Тъй като $\alpha \in (1,2)$, то $log(\alpha) = log_2(\alpha) \in (0,1)$. Тогава имаме

$$b(\underbrace{\alpha-2}_{<0})\alpha^{n-1} + \underbrace{bn - bn^{log(\alpha)} + n}_{\theta(n)} \overset{?}{\leq} 0$$

Ясно е, че асимптотиката се определя от α^{n-1} и че за кое да е $b > 0$ има подходящо n_0. Ако искаме да посочим конкретно n_0 трябва да фиксираме α. Ще го направим за пълнота. Нека $\alpha = \sqrt{2} \in (1,2)$ и заместим в неравенство (3.3)

$$b(\sqrt{2}-2)(\sqrt{2})^{n-1} + bn - b\sqrt{n} + n \overset{?}{\leq} 0$$

От дефиницията на $b = max\{\ldots, 1\} \Rightarrow b \geq 1$

$$b(\sqrt{2}-2)(\sqrt{2})^{n-1} + bn - b\sqrt{n} + n \leq b(\sqrt{2}-2)(\sqrt{2})^{n-1} + bn - b\sqrt{n} + bn \overset{?}{\leq} 0$$

Сега от това, че $b > 0$ можем да разделим двете страни на b

$$(\sqrt{2}-2)(\sqrt{2})^{n-1} + n - \sqrt{n} + n \overset{?}{\leq} 0$$

Ясно е, че $\forall n > 1\,(n \geq \sqrt{n})$

$$(\sqrt{2}-2)(\sqrt{2})^{n-1} + n - \sqrt{n} + n \leq (\sqrt{2}-2)(\sqrt{2})^{n-1} + n \overset{?}{\leq} 0$$

$$(\sqrt{2}-2)(\sqrt{2})^{n-1} + n \overset{?}{\leq} 0$$

Вече е достатъчно просто да забележим, че $n_0 = 9$ ни върши работа

$$\forall n \geq n_0 \left((\sqrt{2}-2)(\sqrt{2})^{n-1} + n \leq 0\right)$$

$$\forall n \geq n_0 \left(b(\sqrt{2}-2)(\sqrt{2})^{n-1} + bn - b\sqrt{n} + n \leq 0\right)$$

което трябваше да докажем.

Тоест доказахме, че $T(n) = O(2^n - (\sqrt{2})^n)$, т.е. $T(n) = O(2^n)$.

- $T(n) = \Omega(2^n)$

  По дефиниция имаме $\Omega(2^n) = \{f \mid \exists c > 0\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0\, (0 \leq c2^n \leq f(n))\}$. Тоест търсим константи $c > 0$ и $n_0 \in \mathbb{N}_0$ такива, че $\forall n \geq n_0\, (c2^n \leq T(n))$.

  Нека $b = \max\left\{\frac{T(0)}{2^0}, \frac{T(1)}{2^1}, \frac{T(2)}{2^2}, \ldots, \frac{T(n_{\text{st}}-1)}{2^{n_{\text{st}}-1}}\right\}$.

  Ще докажем с индукция по n, че за $c = b$ и някое n_0 (което ще установим по-надолу) е изпълнено $\forall n \geq n_0\, (c2^n \leq T(n))$.

  База. Нека $k \in \{0, 1, \ldots, n_{st} - 1\}$. Ще докажем, че $b2^k \leq T(k)$.

  От дефиницията на $b = \min\left\{\frac{T(0)}{2^0}, \frac{T(1)}{2^1}, \frac{T(2)}{2^2}, \ldots, \frac{T(n_{\text{st}}-1)}{2^{n_{\text{st}}-1}}\right\}$ знаем, че $b \leq \frac{T(k)}{2^k}$ откъдето $b2^k \leq T(k)$.

  Индуктивна хипотеза. Нека допуснем, че е изпълнено $\forall m < n\, (b2^m \leq T(m))$.

Индуктивна стъпка. Ще докажем, че е изпълнено за n, тоест че $T(n) \leq b2^n$.

$$T(n) \stackrel{\text{def}}{=} 2T(n-1) + T(log(n)) + n \stackrel{\text{ИХ}}{\geq} 2b2^{n-1} + b2^{log(n)} + n = b2^n + bn + n \stackrel{?}{\geq} b2^n$$

$$b2^n + bn + n \stackrel{?}{\geq} b2^n$$

$$bn + n \stackrel{?}{\geq} 0$$

Очевидно е изпълнено за кое да е $b > 0$ и $n \geq 0$. Оттук си избираме $n_0 = 0$.

Забележка. Обърнете внимание, че този път не добавихме бонус стойност в множеството като образувахме b.. не е нужно винаги!

3.2.3 Полагане

Този метод използваме в комбинация с други методи. Първо полагаме и свеждаме задачата до по-лесна, след което прилагаме друг метод за да намерим асимптотиката.

Задача 3.14. Каква е асимптотиката на $T(n) = 2T(\sqrt{n}) + 1$?

Решение. Полагаме $n = 2^{2^m}$, т.е. $m = log(log(n))$. Нека $S(m) = T(2^{2^m}) = T(n)$. Тогава

$$S(m) = T(2^{2^m}) = 2T(2^{\frac{2^m}{2}}) + 1 = 2T(2^{2^{m-1}}) + 1 = 2S(m-1) + 1$$

$$S(m) = 2S(m-1) + 1$$

Сега използвайки някой от предните методи (а може и някой от бъдещите) доказваме, че $S(m) = \theta(2^m)$. Тоест $T(n) = S(m) = \theta(2^m) = \theta(2^{log(log(n))}) = \theta(log(n))$.

Задача 3.15. Каква е асимптотиката на $T(n) = T(\sqrt{n}) + 1$?

Решение. Полагаме $n = 2^{2^m}$, т.е. $m = log(log(n))$. Нека $S(m) = T(2^{2^m}) = T(n)$. Тогава

$$S(m) = T(2^{2^m}) = T(2^{\frac{2^m}{2}}) + 1 = T(2^{2^{m-1}}) + 1 = S(m-1) + 1$$

$$S(m) = S(m-1) + 1$$

Сега използвайки някой от предните методи (а може и някой от бъдещите) доказваме, че $S(m) = \theta(m)$. Тоест $T(n) = S(m) = \theta(m) = \theta(log(log(n)))$.

Допълнение 3.1: Рекурсивно извикване с корен от аргумента

Нека разгледаме $T(n) = T(\sqrt{n}) + 1$. То извършва константна работа при всяко извикване и извиква $T(\sqrt{n})$. Тоест се интересуваме колко на брой коренувания ще направим преди n да стане 1. Нека вземем едно произволно число - да кажем

30000. Нека сега разгледаме какво се случва когато го коренуваме многократно: $30000 \mapsto 173 \mapsto 13 \mapsto 3 \mapsto 1$. Нека сега запишем числата в двоична бройна система: $111010100110000_{(2)} \mapsto 10101100_{(2)} \mapsto 1101_{(2)} \mapsto 11_{(2)} \mapsto 1_{(2)}$ и да разгледаме тяхната дължина: $15 \mapsto 8 \mapsto 4 \mapsto 2 \mapsto 1$. Може да забележим, че коренуването на дадено число де факто намалява двойно (закръглено нагоре) дължината му в двоична бройна система. Тоест правим $log(\textit{дължината на } n \textit{ в двоична бройна система}) = log(log(n))$ брой стъпки.

3.2.4 Мастър теорема

Макар и силна, не трябва да се подлъгваме по гръмкото име на тази теорема - тя **не** може да намира асимптотиката на какви да е рекурентни уравнения! Въпреки това е много силно пособие, което ще използваме в текущия курс.

Теорема 3.2: Мастър теорема

Нека $a \geq 1, b > 1, f(n)$ - положителна и положим $k = log_b(a)$. Тогава за рекурентното уравнение $T(n) = aT(\frac{n}{b}) + f(n)$ имаме:

$$T(n) = \begin{cases} \theta(n^k) & , \exists \varepsilon > 0 \left(f(n) = O(n^{k-\varepsilon})\right) \\ \theta(n^k log(n)) & , f(n) = \theta(n^k) \\ \theta(f(n)) & , \exists \varepsilon > 0 \left(f(n) = \Omega(n^{k-\varepsilon})\right) \wedge \underbrace{\exists c \in (0,1)\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0 \left(af(\frac{n}{b}) \leq cf(n)\right)}_{\text{условие за регулярност}} \\ \textit{не знаем} & , \text{иначе} \end{cases}$$

Задача 3.16. Каква е асимптотиката на $T(n) = 4T(\frac{n}{2}) + n$?

Решение. Ще използваме мастър теоремата (случай 1):

$$\left| \begin{array}{l} a = 4 \\ b = 2 \\ f(n) = n \\ k = log_b(a) = log_2(4) = 2 \end{array} \right.$$

Нека $\varepsilon = \frac{1}{10}$. Тогава имаме $f(n) = n = O(n^{2-0,1}) = O(n^{1,9}) \overset{MT\,1}{\Longrightarrow} T(n) = \theta(n^2)$.

Задача 3.17. Каква е асимптотиката на $T(n) = 4T(\frac{n}{\sqrt{2}}) + n^3$?

Решение. Ще използваме мастър теоремата (случай 1):

$$\left| \begin{array}{l} a = 4 \\ b = \sqrt{2} \\ f(n) = n^3 \\ k = log_b(a) = log_{\sqrt{2}}(4) = 4 \end{array} \right.$$

Нека $\varepsilon = \frac{1}{10}$. Тогава имаме $f(n) = n = O(n^{4-0,1}) = O(n^{3,9}) \overset{MT\,1}{\Longrightarrow} T(n) = \theta(n^4)$.

Задача 3.18. Каква е асимптотиката на $T(n) = T(\frac{n}{2}) + 1$?

Решение. Ще използваме мастър теоремата (случай 2):

$$\left|\begin{array}{l} a = 1 \\ b = 2 \\ f(n) = 1 \\ k = log_b(a) = log_2(1) = 0 \end{array}\right.$$

Имаме $f(n) = 1 = \theta(1) = \theta(n^0) = \theta(n^k) \overset{MT2}{\Longrightarrow} T(n) = \theta(n^k log(n)) = \theta(log(n))$.

Задача 3.19. Каква е асимптотиката на $T(n) = 2T(\frac{n}{8}) + n$?

Решение. Ще използваме мастър теоремата (случай 3):

$$\left|\begin{array}{l} a = 2 \\ b = 8 \\ f(n) = n \\ k = log_b(a) = log_8(2) = \frac{1}{3} \end{array}\right.$$

Нека $\varepsilon = \frac{1}{6}$. Тогава имаме $f(n) = n = \Omega(n^{\frac{1}{3}+\frac{1}{6}}) = \Omega(\sqrt{n})$. Остана да проверим условието за регулятност: $\exists c \in (0,1)\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0 \left(af(\frac{n}{b}) \leq cf(n)\right)$. Тоест търсим $c \in (0,1)$ и $n_0 \in \mathbb{N}_0$:

$$\forall n \geq n_0 \left(2f\left(\frac{n}{8}\right) \leq cf(n)\right)$$

$$\forall n \geq n_0 \left(2\frac{n}{8} \leq cn\right)$$

Ясно е, че $c = \frac{1}{4}$ и $n_0 = 0$ ни вършат работа. Тоест от МТ 3 имаме, че $T(n) = \theta(f(n)) = \theta(n)$.

Задача 3.20. Каква е асимптотиката на $T(n) = 4T(\frac{n}{2}) + n^2\sqrt{n}$?

Решение. Ще използваме мастър теоремата (случай 3):

$$\left|\begin{array}{l} a = 4 \\ b = 2 \\ f(n) = n^{\frac{5}{2}} \\ k = log_b(a) = log_2(4) = 2 \end{array}\right.$$

Нека $\varepsilon = \frac{1}{10}$. Тогава имаме $f(n) = n^{2,5} = \Omega(n^{2+0,1}) = \Omega(n^{2,1})$. Остана да проверим условието за регулятност: $\exists c \in (0,1)\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0 \left(af(\frac{n}{b}) \leq cf(n)\right)$. Тоест търсим $c \in (0,1)$ и $n_0 \in \mathbb{N}_0$:

$$\forall n \geq n_0 \left(4f\left(\frac{n}{2}\right) \leq cf(n)\right)$$

$$\forall n \geq n_0 \left(4\frac{n^2\sqrt{n}}{4\sqrt{2}} \leq cn^2\sqrt{n}\right)$$

Ясно е, че $c = \frac{1}{\sqrt{2}}$ и $n_0 = 0$ ни вършат работа. Тоест от МТ 3 имаме, че $T(n) = \theta(n^2\sqrt{n})$.

3.2.4.1 Разширение на Мастър теоремата

Макар и да обхваща по-голям набор от функции, тази теорема все още е твърде ограничена. Въпреки това може да улесни решаването на някои задачи многократно.

Теорема 3.3: Мастър теорема (разширена)

Нека $a \geq 1, b > 1, f(n)$ - положителна и положим $k = log_b(a)$. Тогава за рекурентното уравнение $T(n) = aT(\frac{n}{b}) + f(n)$ имаме:

$$T(n) = \begin{cases} \theta(n^k) & , \exists \varepsilon > 0 \left(f(n) = O(n^{k-\varepsilon})\right) \\ \theta(n^k) & , f(n) = \theta(n^k log^{\alpha}(n)) \wedge \alpha < -1 \\ \theta(n^k log^{(2)}(n)) & , f(n) = \theta(n^k log^{\alpha}(n)) \wedge \alpha = -1 \\ \theta(n^k log^{\alpha+1}(n)) & , f(n) = \theta(n^k log^{\alpha}(n)) \wedge \alpha > -1 \\ \theta(f(n)) & , \exists \varepsilon > 0 \left(f(n) = \Omega(n^{k-\varepsilon})\right) \wedge \underbrace{\exists c \in (0,1)\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0 \left(af(\frac{n}{b}) \leq cf(n)\right)}_{\text{условие за регулярност}} \\ \textit{не знаем} & , \text{иначе} \end{cases}$$

където $\alpha \in \mathbb{R}$.

Задача 3.21. Каква е асимптотиката на $T(n) = T(\frac{n}{2}) + \frac{1}{log(n)}$?

Решение. Ще използваме разширената мастър теорема (случай 3):

$$\begin{vmatrix} a = 1 \\ b = 2 \\ f(n) = log^{-1}(n) \\ k = log_b(a) = log_2(1) = 0 \end{vmatrix}$$

Имаме $f(n) = log^{-1}(n) = \theta(n^0 log^{-1}(n)) \overset{EMT\,3}{\Longrightarrow} T(n) = \theta(log(log(n)))$.

Задача 3.22. Каква е асимптотиката на $T(n) = 2T(\frac{n}{4}) + 2\sqrt{n}\, log^3(n)$?

Решение. Ще използваме разширената мастър теорема (случай 4):

$$\begin{vmatrix} a = 2 \\ b = 4 \\ f(n) = 2\sqrt{n}\, log^3(n) \\ k = log_b(a) = log_4(2) = \frac{1}{2} \end{vmatrix}$$

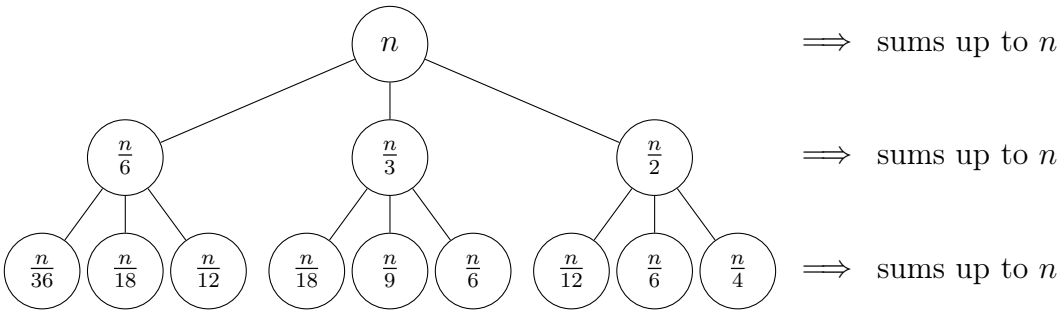
Имаме $f(n) = 2\sqrt{n}\, log^3(n) = \theta(\sqrt{n}\, log^3(n)) \overset{EMT\,4}{\Longrightarrow} T(n) = \theta(\sqrt{n}\, log^4(n))$.

3.2.5 Дърво на рекурсия

Аналогично на метода с развиване, текущия метод не е формален. Той служи само за придобиване на интуиция. Необходимо е формално доказателство с индукция след като се заподозрее асимптотиката.

Задача 3.23. Каква е асимптотиката на $T(n) = T\left(\frac{n}{6}\right) + T\left(\frac{n}{3}\right) + T\left(\frac{n}{2}\right) + n$?

Решение. Нека разгледаме дървото на рекурсия на $T(n)$:



Може да забележим, че на всяко ниво сбора на нехомогенните части дава точно n. Освен това дървото е пълно до ниво $\lceil log_6(n) \rceil$[3] (при база $n = 1$), съответно дървото е с височина $\lceil log_2(n) \rceil$[3]. Тоест заподозряхме, че $T(n) = \theta(n\, log_2(n)) + O\big(n(log_6(n) - log_2(n))\big) = \theta(n\, log(n))$. Сега ще го докажем формално. Нека $n_{st} \in \mathbb{N}_0$ е достатъчно голямо.

- $T(n) = O(n\, log(n))$

  По дефиниция имаме $O(n\, log(n)) = \{f \mid \exists c > 0\, \exists n_0 \in \mathbb{N}_0\, \forall n \geq n_0\, (0 \leq f(n) \leq c.n\, log(n))\}$. Тоест търсим $c > 0$ и $n_0 \in \mathbb{N}_0$ такива, че $\forall n \geq n_0\, (T(n) \leq n\, log(n))$.

  Нека $b = \max\left\{\frac{T(2)}{2log(2)}, \frac{T(3)}{3log(3)}, \ldots, \frac{T(n_{\text{st}}-1)}{(n_{\text{st}}-1)log(n_{\text{st}}-1)}, 1\right\}$. (виж **Забележка 3.1**)

  Ще докажем с индукция по n, че за $c = b$ и някое n_0 (което ще установим по-надолу) е изпълнено $\forall n \geq n_0\, (T(n) \leq n\, log(n))$.

  База. Нека $k \in \{2, 3, \ldots, n_{st} - 1\}$. Ще докажем, че $T(k) \leq b.k.log(k)$.

  От дефиницията на $b = \max\left\{\frac{T(2)}{2log(2)}, \frac{T(3)}{3log(3)}, \ldots, \frac{T(n_{\text{st}}-1)}{(n_{\text{st}}-1)log(n_{\text{st}}-1)}, 1\right\}$ знаем, че $b \geq \frac{T(k)}{k.log(k)}$ откъдето $b.k.log(k) \geq T(k)$ или още $T(k) \leq b.k.log(k)$.

  Индуктивна хипотеза. Нека допуснем, че е изпълнено $\forall m < n\, (T(m) \leq b.m.log(m))$.

  Индуктивна стъпка. Ще докажем, че е изпълнено за n, тоест че $T(n) \leq b.n.log(n)$.

$$T(n) \overset{\text{def}}{=} T\left(\frac{n}{6}\right) + T\left(\frac{n}{3}\right) + T\left(\frac{n}{2}\right) + n \overset{\text{ИХ}}{\leq} \frac{bn}{6} log\left(\frac{n}{6}\right) + \frac{bn}{3} log\left(\frac{n}{3}\right) + \frac{bn}{2} log\left(\frac{n}{2}\right) + n \overset{?}{\leq} b.n.log(n)$$

$$\frac{bn}{6}\big(log(n) - log(6)\big) + \frac{bn}{3}\big(log(n) - log(3)\big) + \frac{bn}{2}\big(log(n) - log(2)\big) + n \overset{?}{\leq} b.n.log(n)$$

$$b.n.log(n) + \left(1 - \frac{b.log(6)}{6} - \frac{b.log(3)}{3} - \frac{b.log(2)}{2}\right) n \overset{?}{\leq} b.n.log(n)$$

[3] Може да докажем тези две твърдения с индукция и да докажем формално сумата от всички върхове, че е в интервала $[n\, log_6(n), n\, log_2(n)]$, откъдето да получим асимптотиката $\theta(n\, log(n))$. В текущия курс няма да правим индукции от такъв характер.

$$\left(1 - \frac{b.log(6)}{6} - \frac{b.log(3)}{3} - \frac{b.log(2)}{2}\right) n \overset{?}{\leq} 0$$

Може да забележим, че при $\frac{b.log(6)}{6} + \frac{b.log(3)}{3} + \frac{b.log(2)}{2} \geq 1$ горното неравенство ще е изпълнено за всяко $n \geq 1$. Оттук си избираме $n_0 = 2$ (при $n_0 = 1$ имаме $\frac{T(1)}{1log(1)} = \frac{T(1)}{0}$).

Остана да видим кога е изпълнено $\frac{b.log(6)}{6} + \frac{b.log(3)}{3} + \frac{b.log(2)}{2} \geq 1$. Очевидно е изпълнено когато $b \geq 1$. Това сме си го подсигурили от дефиницията на $b = \max\{\ldots, 1\} \Rightarrow b \geq 1$.

Тоест доказахме, че за $c = b \wedge n_0 = 1$ е изпълнено $\forall n \geq n_0 \, (0 \leq T(n) \leq c.n.log(n))$. Казано с други думи $T(n) = O(n\,log(n))$.

- $T(n) = \Omega(n\,log(n))$ - Докажете за упражнение.

3.2.6 Теорема на Akra-Bazzi

Сега ще разгледаме метод, който е строго по-силен от Мастър теоремата, която е твърде ограничена - има една единствена поява вдясно. Теоремата на Akra-Bazzi позволява решаването на частен случай рекурентни уравнения с една или повече появи вдясно.

Теорема 3.4: Akra-Bazzi

Нека

$$T(n) = \begin{cases} \theta(1) & , 1 \leq n \leq n_0 \\ a_1 T(b_1 n) + \cdots + a_k T(b_k n) + f(n) & , n > n_0 \end{cases}$$

където

1. $1 \leq n \in \mathbb{R}$
2. $\forall i \in \{1, \ldots, k\} \left(n_0 \geq \frac{1}{b_i} \wedge n_0 \geq \frac{1}{1-b_i}\right)$
3. $\forall i \in \{1, \ldots, k\} \left(a_i > 0\right)$
4. $\forall i \in \{1, \ldots, k\} \left(b_i \in (0, 1)\right)$
5. $k \in \mathbb{N}^+$
6. $f(n)$ е неотрицателна функция, удволетворяваща условието за полиномиално нарастване
7. p е уникално число, за което $\sum_{i=1}^{k} a_i b_i^p = 1$

Тогава

$$T(n) \asymp n^p \left(1 + \int_1^n \frac{f(t)}{t^{p+1}} dt\right)$$

Дефиниция 3.1: Условие за полиномиално нарастване (в контекста на Теорема 3.4)

Ще казваме, че $f(n)$ удволетворява условието за полиномиално нарастване, т.с.т.к. $\exists c_1 > 0 \, \exists c_2 > 0 \, \forall n \geq 1 \, \forall i \in \{1, \ldots, k\} \, \forall t \in [b_i n, n] \big(c_1 f(n) \leq f(t) \leq c_2 f(n)\big)$.

Задача 3.24. Каква е асимптотиката на $T(n) = \frac{1}{4}T\left(\frac{n}{4}\right) + \frac{3}{4}T\left(\frac{3n}{4}\right) + 1$?

Решение. Ще покажем, че седемте условия от теоремата на Akra-Bazzi са изпълнени:

1. Това условие няма какво да го проверяваме.
2. Проверяваме, че $n_0 = 4$ ни върши работа: $\forall i \in \{1,2\}\left(4 \geq \frac{1}{b_i} \wedge 4 \geq \frac{1}{1-b_i}\right)$.
3. Директно проверяваме $\forall i \in \{1,2\}\left(a_i > 0\right)$.
4. Директно проверяваме $\forall i \in \{1,2\}\left(b_i \in (0,1)\right)$.
5. Това условие няма какво да го проверяваме.
6. Проверяваме, че $c_1 = c_2 = 1$ ни вършат работа: $\forall n \geq 1\, \forall i \in \{1,2\}\, \forall t \in [b_i n, n]\left(1 \leq 1 \leq 1\right)$.
7. Проверяваме, че $p = 0$ ни върши работа: $\frac{1}{4}\left(\frac{1}{4}\right)^0 + \frac{3}{4}\left(\frac{3}{4}\right)^0 = 1$.

Тогава прилагаме теоремата на Akra-Bazzi и получаваме $T(n) \asymp n^0\left(1 + \int_1^n \frac{1}{t}dt\right) = \theta(log(n))$.

3.2.7 Задачи

Основната идея за строене на рекурентно уравнение за сложността на рекурсивен алгоритъм е за всяко рекурсивно извикване с параметър k да добавим отдясно на равенството $T(k)$. Останалата *работа* ще представлява нехомогенната част на рекурентното уравнение.

Задача 3.25. Даден е следния алгоритъм:

```
1. Func(n) : // n ∈ ℕ⁺
2. |  s ← 0;
3. |  for i ← 1 to n - 1
4. |  |  s ← s + 2 * Func(i) + 1;
5. |  return s;
```

Каква е сложността му по време (спрямо големината на входа n)?

Решение. Имаме $n-1$ на брой рекурсивни извиквания.. това означава, че за всяко едно от тях трябва да добавим отдясно на равенството $T(\textit{подходящ параметър})$. Останалата *работа* в случая е инициализацията на s, инкрементацията на i и актуализацията на s, т.е. $\theta(n)$. Тоест рекурентното уравнение е $T(n) = T(n-1) + T(n-2) + \cdots + T(1) + \theta(n)$. Вече намерихме асимптотиката на подобно рекурентно уравнение в **Задача 3.8**.

$$\left|\begin{array}{l} T(n) = T(n-1) + T(n-2) + \cdots + T(1) + \theta(n) \\ T(n-1) = T(n-2) + T(n-3) + \cdots + T(1) + \theta(n-1) \end{array}\right.$$

Тогава $T(n) - T(n-1) = T(n-1) + \theta(n) - \theta(n-1)$ или още $T(n) = 2T(n-1) + \theta(1)$[4]. Вече може да го решим чрез метода с характеристично уравнение.

[4] В общия случай получаваме произволна функция (та може дори асимптотично отрицателна).. преценете защо тук знаем, че имаме $\theta(1)$.

- (Хомогенна част)
  $x = 2 \ \mapsto \ \{2\}_M$
- (Нехомогенна част)
  $\theta(1) \ \mapsto \ \{1\}_M$

Оттук получаваме, че $T(n) = c_1 1^n + c_2 2^n = \theta(2^n)$.

Задача 3.26. Даден е следния алгоритъм:

```
1. Func(n) : // n ∈ ℕ⁺
2.     s ← 0;
3.     for i ← 1 to n - 1
4.         s ← s + Func(i) + Func(i) + 1;
5.     return s;
```

Каква е сложността му по време (спрямо големината на входа n)?

Решение. Имаме $2(n-1)$ на брой рекурсивни извиквания.. това означава, че за всяко едно от тях трябва да добавим отдясно на равенството $T(\textit{подходящ параметър})$. Останалата *работа* в случая е инициализацията на s, инкрементацията на i и актуализацията на s, т.е. $\theta(n)$. Тоест рекурентното уравнение е $T(n) = 2T(n-1) + 2T(n-2) + \cdots + 2T(1) + \theta(n)$.

$$\left| \begin{array}{l} T(n) = 2T(n-1) + 2T(n-2) + \cdots + 2T(1) + \theta(n) \\ T(n-1) = 2T(n-2) + 2T(n-3) + \cdots + 2T(1) + \theta(n-1) \end{array} \right.$$

Тогава $T(n) - T(n-1) = 2T(n-1) + \theta(n) - \theta(n-1)$ или още $T(n) = 3T(n-1) + \theta(1)$[4]. Вече може да го решим чрез метода с характеристично уравнение.

- (Хомогенна част)
  $x = 3 \ \mapsto \ \{3\}_M$
- (Нехомогенна част)
  $\theta(1) \ \mapsto \ \{1\}_M$

Оттук получаваме, че $T(n) = c_1 1^n + c_2 3^n = \theta(3^n)$.