Рекурсия

Трифон Трифонов

Увод в програмирането,
спец. Компютърни науки, 1 поток, 2018/19 г.

3 януари 2019 г.

Какво е рекурсия?

Какво е рекурсия?

N. Wirth, Algorithms and Data Structures, Fig 3.1

Какво е рекурсия?

Какво е рекурсия?

- Повторение чрез позоваване на себе си

Какво е рекурсия?

- Повторение чрез позоваване на себе си
- “приятелите на моите приятели са и мои приятели”

Какво е рекурсия?

- Повторение чрез позоваване на себе си
- “приятелите на моите приятели са и мои приятели”
- директориите съдържат файлове и директории

Какво е рекурсия?

- Повторение чрез позоваване на себе си
- "приятелите на моите приятели са и мои приятели"
- директориите съдържат файлове и директории
- PHP = PHP Hypertext preprocessor

Какво е рекурсия?

- Повторение чрез позоваване на себе си
- "приятелите на моите приятели са и мои приятели"
- директориите съдържат файлове и директории
- PHP = PHP Hypertext preprocessor
- за да строшите камък:

Какво е рекурсия?

- Повторение чрез позоваване на себе си
- “приятелите на моите приятели са и мои приятели”
- директориите съдържат файлове и директории
- PHP = PHP Hypertext preprocessor
- за да строшите камък:
    - ударете с чука, за да натрошите камъка на части

Какво е рекурсия?

- Повторение чрез позоваване на себе си
- "приятелите на моите приятели са и мои приятели"
- директориите съдържат файлове и директории
- PHP = PHP Hypertext preprocessor
- за да строшите камък:
  - ударете с чука, за да натрошите камъка на части
  - строшете получените по-малки камъни

Какво е рекурсия?

- Повторение чрез позоваване на себе си
- "приятелите на моите приятели са и мои приятели"
- директориите съдържат файлове и директории
- PHP = PHP Hypertext preprocessor
- за да строшите камък:
  - ударете с чука, за да натрошите камъка на части
  - строшете получените по-малки камъни
- за да разберете какво е рекурсия, трябва да разберете какво е рекурсия

Рекурсията в математиката

$$n! = \begin{cases} 1, & n = 0, \\ n(n-1)!, & n > 0. \end{cases}$$

$$n! = 1.2\ldots n = \prod_{i=1}^{n} i$$

$$4! = 4.3! = 4.3.2! = 4.3.2.1! = 4.3.2.1.\underbrace{0!}_{1}$$

$$= 24$$

Рекурсията в математиката

$$n! = \begin{cases} 1, & n = 0, \\ n(n-1)!, & n > 0. \end{cases}$$

$$x^n = \begin{cases} 1, & n = 0, \\ x.x^{n-1}, & n > 0, \\ \frac{1}{x^{-n}}, & n < 0. \end{cases}$$

$$3^{-2} = \frac{1}{3^2} = \frac{1}{3.3^1} = \frac{1}{3.3.3^0} = \frac{1}{3.3.1} = \frac{1}{9}$$

Рекурсията в математиката

$$n! = \begin{cases} 1, & n = 0, \\ n(n-1)!, & n > 0. \end{cases}$$

$$x^n = \begin{cases} 1, & n = 0, \\ x.x^{n-1}, & n > 0, \\ \frac{1}{x^{-n}}, & n < 0. \end{cases}$$

$$gcd(a, b) = \begin{cases} a, & a = b, \\ gcd(a-b, b), & a > b, \\ gcd(a, b-a), & a < b. \end{cases}$$

$a, b > 0$

Рекурсията в математиката

$G \subseteq \mathbb{N}^2$

$G = \{(0,0)\}$

$$n! = \begin{cases} 1, & n = 0, \\ n(n-1)!, & n > 0. \end{cases}$$

$f(x) = x$

$f(5) = f(6) + 1$

$= f(7) + 2$

$= f(8) + 3$

$= \ldots$

$= \ldots$

$$x^n = \begin{cases} 1, & n = 0, \\ x.x^{n-1}, & n > 0, \\ \frac{1}{x^{-n}}, & n < 0. \end{cases}$$

$f(x) = \begin{cases} x+1, & x>0 \\ 0, & x=0 \end{cases}$

$$gcd(a,b) = \begin{cases} a, & a = b, \\ gcd(a-b,b), & a > b, \\ gcd(a,b-a), & a < b. \end{cases}$$

$f(x) = \begin{cases} x+c, & x>0 \\ 0, & x=0 \end{cases}$

$$f(x) = \begin{cases} 0, & x = 0, \\ f(x+1) - 1, & x > 0. \end{cases}$$

$x+1-1$

$f(x) = \begin{cases} 0, & x=0 \\ \text{недеф.}, & x>0 \end{cases}$

Как се решават задачи с рекурсия?

- Декомпозиция — свеждане на дадена задача към множество от по-прости задачи

Как се решават задачи с рекурсия?

- Декомпозиция — свеждане на дадена задача към множество от по-прости задачи
- Рекурсията е вид декомпозиция, при който свеждаме задача към множество от по-прости задачи подобни на първоначалната

Как се решават задачи с рекурсия?

- Декомпозиция — свеждане на дадена задача към множество от по-прости задачи
- Рекурсията е вид декомпозиция, при който свеждаме задача към множество от по-прости задачи подобни на първоначалната
- Как работи:

Как се решават задачи с рекурсия?

- Декомпозиция — свеждане на дадена задача към множество от по-прости задачи
- Рекурсията е вид декомпозиция, при който свеждаме задача към множество от по-прости задачи подобни на първоначалната
- Как работи:
  - Показваме решението на най-простите задачи (база, дъно)

Как се решават задачи с рекурсия?

- Декомпозиция — свеждане на дадена задача към множество от по-прости задачи
- Рекурсията е вид декомпозиция, при който свеждаме задача към множество от по-прости задачи подобни на първоначалната
- Как работи:
  - Показваме решението на най-простите задачи (база, дъно)
  - Показваме как по-сложна задача се свежда към една или няколко по-прости (стъпка)

Математическа индукция

Дефиниция

Математическата индукция е метод за доказателство, използващ като предпоставка свойството, което се доказва.

Математическа индукция

Дефиниция

Математическата индукция е метод за доказателство, използващ като предпоставка свойството, което се доказва.

Пример: Да се докаже, че $2 + 4 + ... + 2n = n(n + 1)$.

Математическа индукция

Дефиниция

Математическата индукция е метод за доказателство, използващ като предпоставка свойството, което се доказва.

Пример: Да се докаже, че $2 + 4 + ... + 2n = n(n + 1)$.

Доказателство:

- за $n = 0$: трябва да проверим, че $0 = 0.1$ ✓

Математическа индукция

Дефиниция

Математическата индукция е метод за доказателство, използващ като предпоставка свойството, което се доказва.

Пример: Да се докаже, че $2 + 4 + \ldots + 2n = n(n+1)$.

Доказателство:

- за $n = 0$: трябва да проверим, че $0 = 0.1 \quad \checkmark$
- нека допуснем, че сме доказали свойството за дадено n

Математическа индукция

Дефиниция

Математическата индукция е метод за доказателство, използващ като предпоставка свойството, което се доказва.

Пример: Да се докаже, че $2 + 4 + \ldots + 2n = n(n + 1)$.

Доказателство:

- за $n = 0$: трябва да проверим, че $0 = 0.1$ ✓
- нека допуснем, че сме доказали свойството за дадено n
- ще го докажем за $n + 1$:

Математическа индукция

Дефиниция

Математическата индукция е метод за доказателство, използващ като предпоставка свойството, което се доказва.

Пример: Да се докаже, че $2 + 4 + \ldots + 2n = n(n + 1)$.

Доказателство:

- за $n = 0$: трябва да проверим, че $0 = 0.1$ ✓
- нека допуснем, че сме доказали свойството за дадено n
- ще го докажем за $n + 1$:
- $(2 + 4 + \ldots + 2n) + 2(n + 1) = n(n + 1) + 2(n + 1) = (n + 1)(n + 2)$ ✓

Математическа индукция

Дефиниция

Математическата индукция е метод за доказателство, използващ като предпоставка свойството, което се доказва.

Пример: Да се докаже, че $2 + 4 + \ldots + 2n = n(n + 1)$.

Доказателство:

- за $n = 0$: трябва да проверим, че $0 = 0.1$ ✓
- нека допуснем, че сме доказали свойството за дадено n
- ще го докажем за $n + 1$:
- $(2+4+\ldots+2n)+2(n+1) = n(n+1)+2(n+1) = (n+1)(n+2)$ ✓
- **Следователно:** доказахме свойството за произволно n. □

Математическа индукция

Дефиниция

Математическата индукция е метод за доказателство, използващ като предпоставка свойството, което се доказва.

Пример: Да се докаже, че $2 + 4 + \ldots + 2n = n(n + 1)$.

Доказателство:

- за $n = 0$: трябва да проверим, че $0 = 0.1$ $\checkmark$
- нека допуснем, че сме доказали свойството за дадено n
- ще го докажем за $n + 1$:
- $(2+4+\ldots+2n)+2(n+1) = n(n+1)+2(n+1) = (n+1)(n+2)$ $\checkmark$
- **Следователно:** доказахме свойството за произволно n. $\square$

Математическата индукция е рекурсивен метод за доказателство.

Рекурсията в програмирането

Дефиниция

Рекурсивна функция наричаме функция, която извиква себе си пряко или косвено.

Рекурсията в програмирането

Дефиниция

Рекурсивна функция наричаме функция, която извиква себе си пряко или косвено.

Рекурсивни функции се поддържат от почти всички съвременни езици за програмиране.

Рекурсията в програмирането

Дефиниция

Рекурсивна функция наричаме функция, която извиква себе си пряко или косвено.

Рекурсивни функции се поддържат от почти всички съвременни езици за програмиране.

Теорема

Всяка програма с цикли може да се напише с рекурсия и обратно.

Примери за рекурсивни функции

Да се напише функция, която пресмята рекурсивно:

1. $n!$

Стекови рамки на рекурсивни функции

main	n	4	`cout << fact(4);`

Стекови рамки на рекурсивни функции

fact		адрес на връщане	`return 4 * fact(3);`
	n	4	
main	n	4	`cout << fact(4);`

Стекови рамки на рекурсивни функции

fact		адрес на връщане	`return 3 * fact(2);`
	n	3	
fact		адрес на връщане	`return 4 * fact(3);`
	n	4	
main	n	4	`cout << fact(4);`

Стекови рамки на рекурсивни функции

fact		адрес на връщане	`return 2 * fact(1);`
	n	2	
fact		адрес на връщане	`return 3 * fact(2);`
	n	3	
fact		адрес на връщане	`return 4 * fact(3);`
	n	4	
main	n	4	`cout << fact(4);`

Стекови рамки на рекурсивни функции

fact		адрес на връщане	`return 1 * fact(0);`
	n	1	
fact		адрес на връщане	`return 2 * fact(1);`
	n	2	
fact		адрес на връщане	`return 3 * fact(2);`
	n	3	
fact		адрес на връщане	`return 4 * fact(3);`
	n	4	
main	n	4	`cout << fact(4);`

Стекови рамки на рекурсивни функции

`fact`		адрес на връщане	`return 1;`
	`n`	0	
`fact`		адрес на връщане	`return 1 * fact(0);`
	`n`	1	
`fact`		адрес на връщане	`return 2 * fact(1);`
	`n`	2	
`fact`		адрес на връщане	`return 3 * fact(2);`
	`n`	3	
`fact`		адрес на връщане	`return 4 * fact(3);`
	`n`	4	
`main`	`n`	4	`cout << fact(4);`

Стекови рамки на рекурсивни функции

fact		адрес на връщане	`return 1 * 1;`
	n	1	
fact		адрес на връщане	`return 2 * fact(1);`
	n	2	
fact		адрес на връщане	`return 3 * fact(2);`
	n	3	
fact		адрес на връщане	`return 4 * fact(3);`
	n	4	
main	n	4	`cout << fact(4);`

Стекови рамки на рекурсивни функции

fact		адрес на връщане	`return 2 * 1;`
	n	2	
fact		адрес на връщане	`return 3 * fact(2);`
	n	3	
fact		адрес на връщане	`return 4 * fact(3);`
	n	4	
main	n	4	`cout << fact(4);`

Стекови рамки на рекурсивни функции

fact		адрес на връщане	`return 3 * 2;`
	n	3	
fact		адрес на връщане	`return 4 * fact(3);`
	n	4	
main	n	4	`cout << fact(4);`

Стекови рамки на рекурсивни функции

fact		адрес на връщане	`return 4 * 6;`
	n	4	
main	n	4	`cout << fact(4);`

Стекови рамки на рекурсивни функции

main	n	4	cout << 24;

factd(1000) $\longrightarrow$ x clk.

factr(1000) $\longrightarrow$ y clk.

y > x

factc(2000) $\longrightarrow$ 2x clk

factr(2000) $\longrightarrow$ 2y clk.

Примери за рекурсивни функции

Да се напише функция, която пресмята рекурсивно:

1. $n!$
2. НОД

Примери за рекурсивни функции

Да се напише функция, която пресмята рекурсивно:

1. $n!$
2. НОД
3. x^n

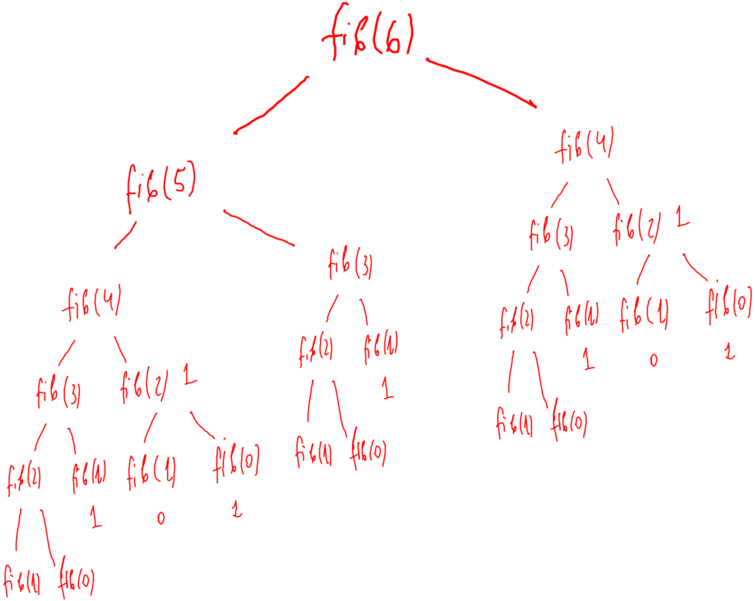
Примери за рекурсивни функции

Да се напише функция, която пресмята рекурсивно:

1. $n!$
2. НОД
3. x^n
4. числата на Фибоначи

0, 1, 1, 2, 3, 5, 8, 13

$$f_n = \begin{cases} 0, & n=0 \\ 1, & n=1 \\ f_{n-1}+f_{n-2}, & n>1 \end{cases}$$


fib(6)
fib(5)
fib(4)
fib(4)
fib(3)
fib(3)
fib(2)
fib(3)
fib(2)
1
fib(2)
fib(1)
fib(2)
fib(1)
fib(1)
fib(0)
1
1
0
1
fib(2)
fib(1)
fib(1)
fib(0)
fib(1)
fib(0)
1
0
1
fib(1)
fib(0)
1
fib(1)
fib(0)

Примери за рекурсивни функции

Да се напише функция, която пресмята рекурсивно:

1. $n!$
2. НОД
3. x^n
4. числата на Фибоначи
5. числата на Фибоначи, но по-бързо.

Примери за рекурсивни функции

Да се напише функция, която пресмята рекурсивно:

1. $n!$
2. НОД
3. x^n
4. числата на Фибоначи
5. числата на Фибоначи, но по-бързо.
6. <израз> със скоби, където
   - <израз> ::= <цифра> | (<израз><операция><израз>)
   - <цифра> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
   - <операция> ::= + | - | * | /

((2 − (2*3)+(4*7))*(5+2))

0

$(2+3)$

$\uparrow$

5