Рекурсия

Трифон Трифонов

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

3 януари 2019 г.

Какво е рекурсия?

N. Wirth, Algorithms and Data Structures, Fig 3.1

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

Рекурсията в математиката

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

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

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

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

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.

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. □

Математическата индукция е рекурсивен метод за доказателство.

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. $n!$
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> ::= + | - | * | /

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);`

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> ::= + | - | * | /