



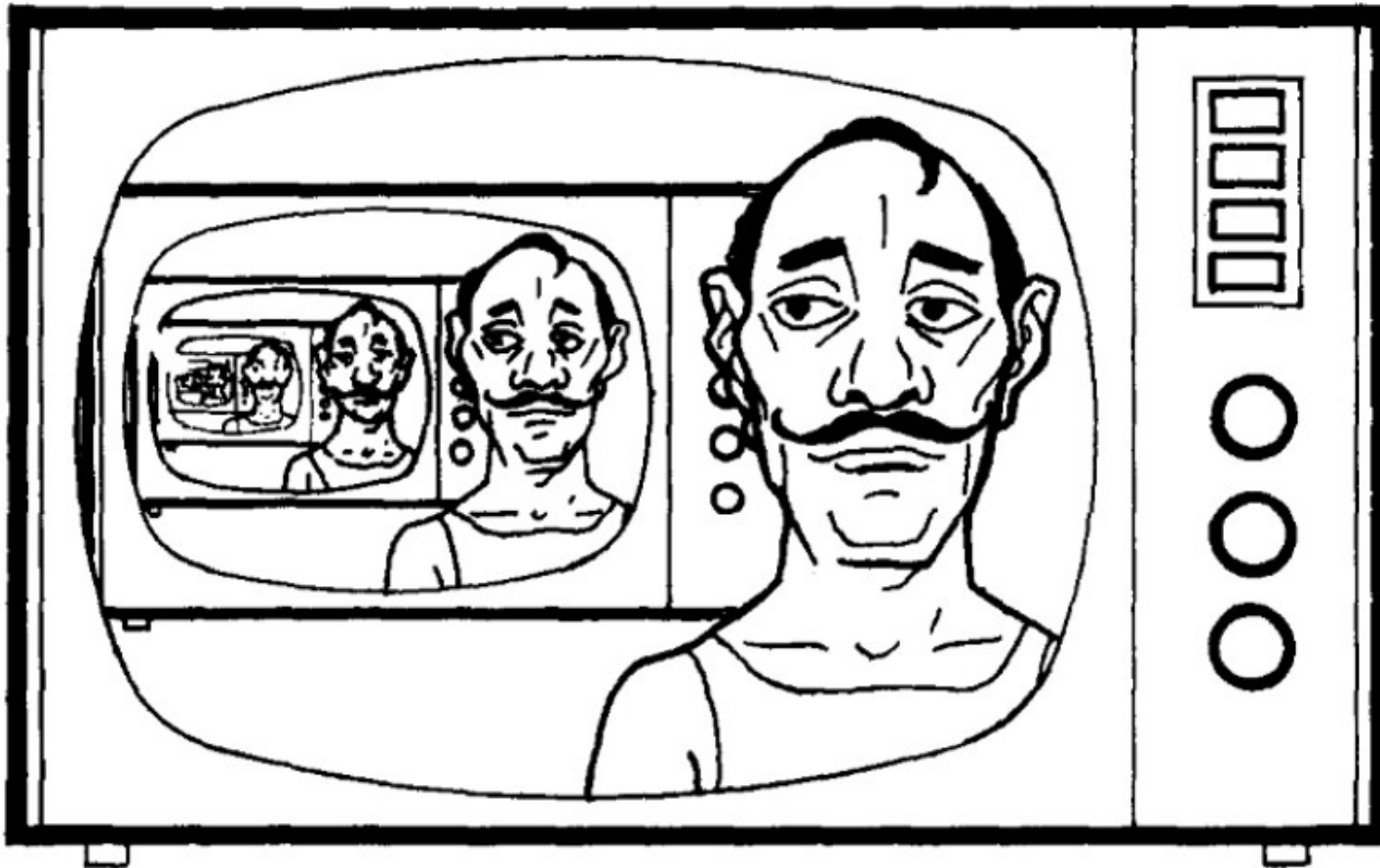
Рекурсия

Какво е рекурсия?

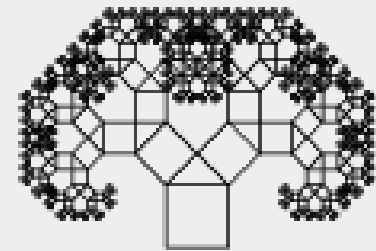
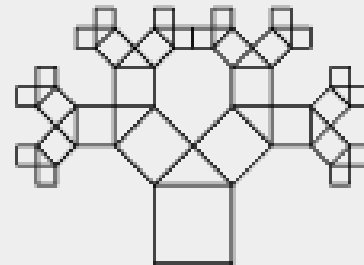
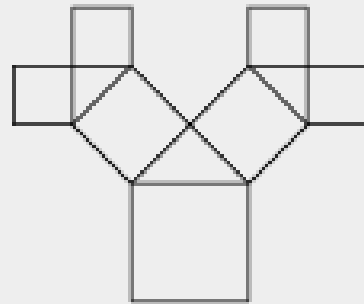
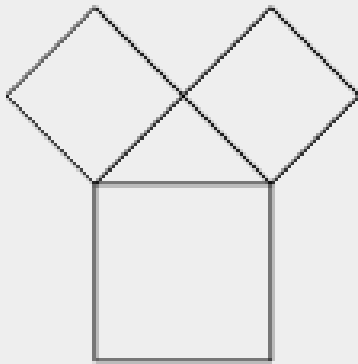
Какво е рекурсия?



Какво е рекурсия?



Какво е рекурсия?



Какво е рекурсия?

- Повторение чрез позоваване на себе си
- Рекурсивни примери:
 - приятелите на моите приятели са и мои приятели
 - директориите съдържат файлове и директории
 - PHP = PHP Hypertext Preprocessor
 - за да строшите камък, ударете с чука и строшете по-малките камъни
 - за да разберете рекурсията, трябва да разберете рекурсията

Рекурсия в математиката

$$n! = \begin{cases} 1, & n=0 \\ n(n-1)!, & n>0 \end{cases} \quad x^n = \begin{cases} 1, & n=0 \\ x \cdot x^{n-1}, & n>0 \\ \frac{1}{x^{-n}}, & n<0 \end{cases}$$

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

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

За какво служи рекурсията?

- За решаването на дадена задача:
 - показва се решението на най-простите задачи (база, дъно)
 - показва се как се свежда сложна задача към една или няколко по-прости (стъпка)

Индукция

- Метод за доказателство, използващ като предпоставка свойството, което се доказва
- Пример: $2 + 4 + \dots + 2n = n(n+1)$
 - за $n = 0$ — $0 = 0.1$ — вярно
 - нека е вярно за n
 - $(2 + 4 + \dots + 2n) + 2(n + 1) =$
 $= n(n+1) + 2(n + 1) = (n+1)(n+2)$

Рекурсия в програмирането

- Функция, която извиква себе си пряко или косвено
- Рекурсията се поддържа от почти всички съвременни езици за програмиране

Примери за рекурсивни функции

- Факториел
- НОД
- Степен
- Числа на Фибоначи
 - линейна и дървовидна рекурсия
 - мемоизация
- Израз със скоби

Рекурсия vs цикли

- Теорема: всяка програма с цикли може да се напише с рекурсия и обратно.

Рекурсивни функции за масиви

- намиране на сума
- проверка за съществуване на елемент
- проверка за монотонно нарастване
- проверка за различни елементи

Търсене с връщане назад (backtracking)

- Лесно: имаме ясна последователност, в която да обработим всички случаи
- Трудно: последователността на обработка не е ясна предварително

Търсене с връщане назад (backtracking)

- Проба и грешка:
 - ако имаме няколко варианта как да продължим: избираме произволно (стъпка напред, проба)
 - когато се окажем без никакъв избор се връщаме и коригираме последния направен избор (стъпка назад, грешка)
 - когато получим желания резултат: успех!
 - ако се върнем в началото: провал!

Задачи за търсене с връщане

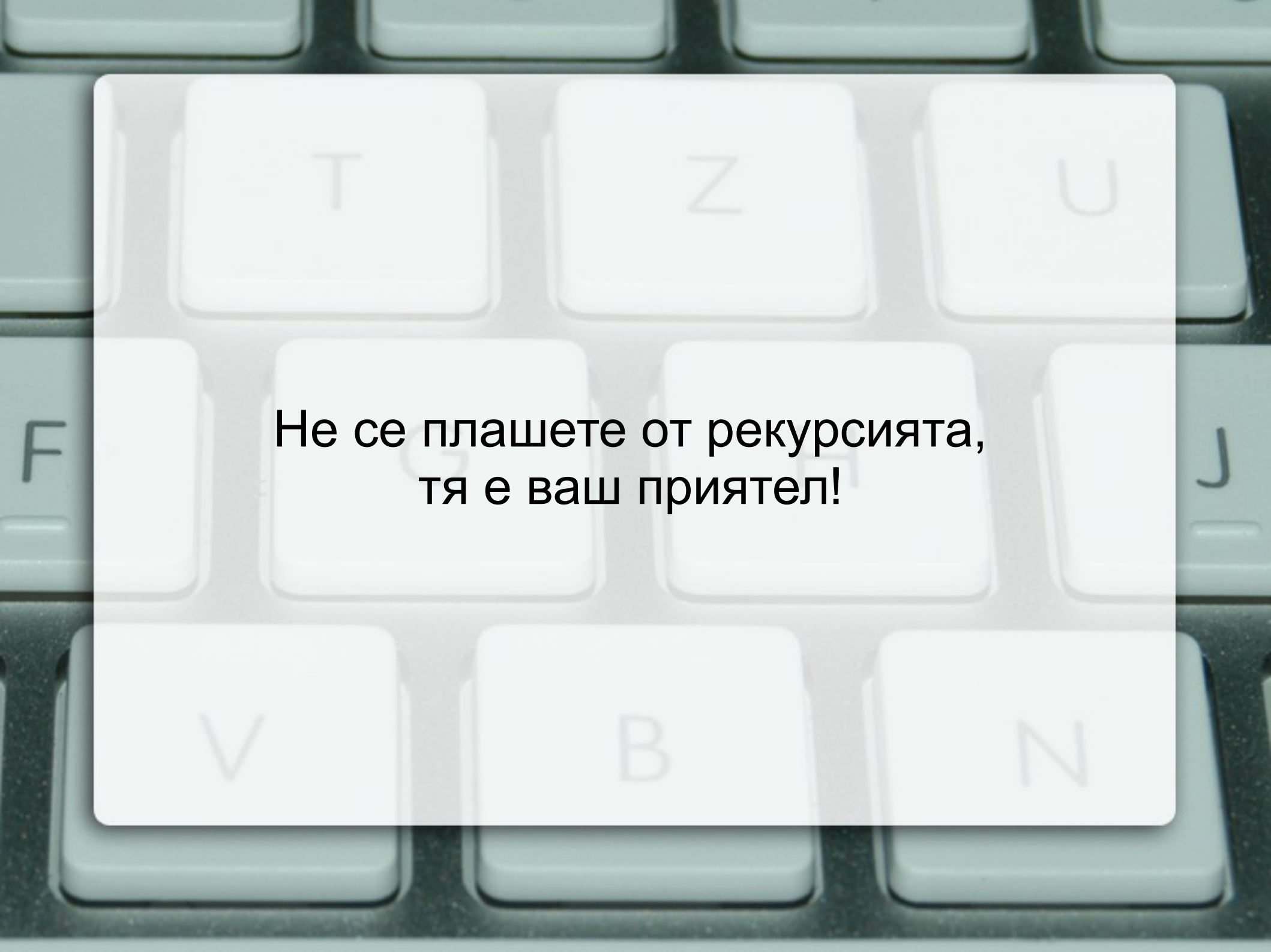
- Търсене на път в лабиринт
- Търсене на път между градове

Предимства на рекурсията

- Добра изразителност
- Хубави математически свойства
- Удобство при решаването на рекурсивно дефинирани задачи
- Удобна за реализиране за търсене с връщане назад (backtracking)
- Удобна за алгоритми от тип “разделяй и владей”

Недостатъци на рекурсията

- Скрито използване на памет за стекови рамки
- При неправилно използване може да е неефективна
- Понякога има нужда от помощни функции
- Изглежда плашеща за много програмисти :)



Не се плашете от рекурсията,
тя е ваш приятел!