

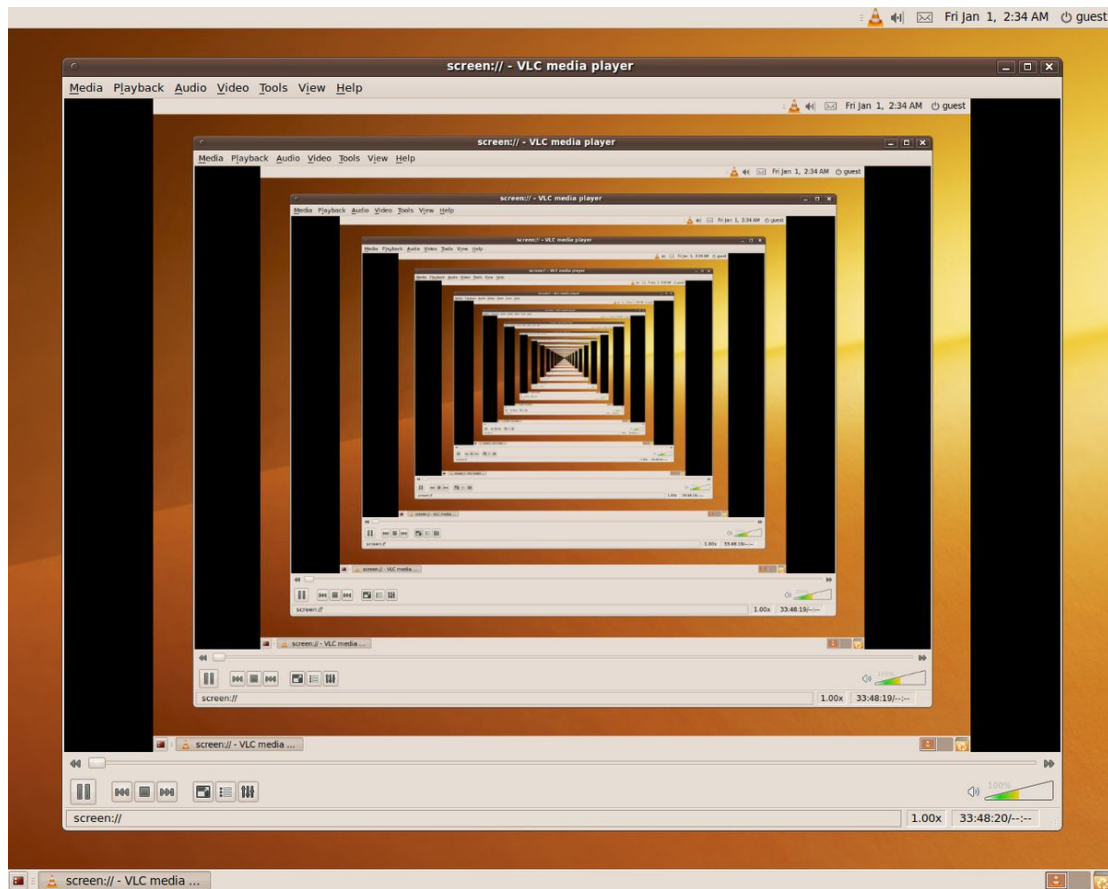
Рекурсия

УП практикум, 2ра група
Богомил Стоянов
Виолета Кастрева

Какво е рекурсия?

Функция извиква себе си с опростени в някакъв смисъл параметри до достигането на базов случай.

Разбиване на проблема на по-лесно решими подпроблеми




Базов случай

- Най-простия случай на една функция
- Винаги трябва да го има
- Връща стойност директно без да извиква себе си

```
1 // factorial base case
2 if (n == 0 || n == 1) {
3     return 1;
4 }
5
6 // fibonacci base case
7 if (n == 0) {
8     return 0;
9 } else if (n == 1) {
10    return 1;
11 }
12
13 // binary search base case
14 if (low > high) {
15     return false;
16 }
```

Рекурсивно извикване

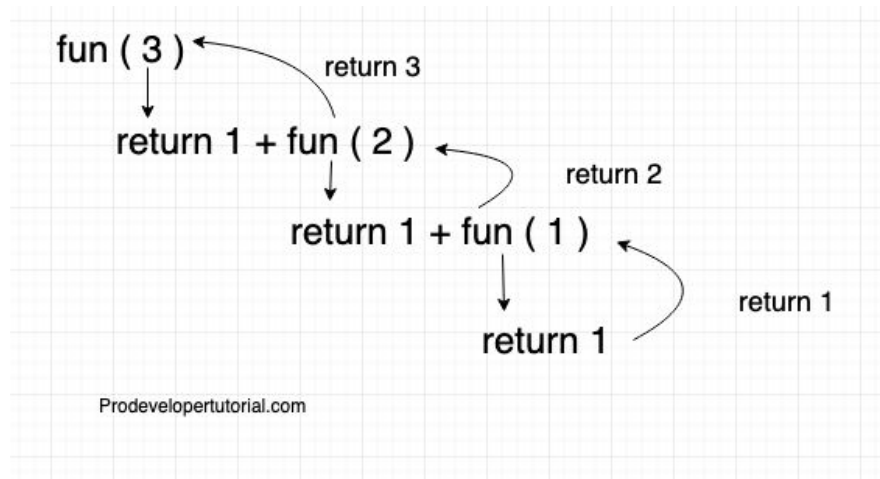
Дефинира как проблемът се
разбива на по-прости
подпроблеми



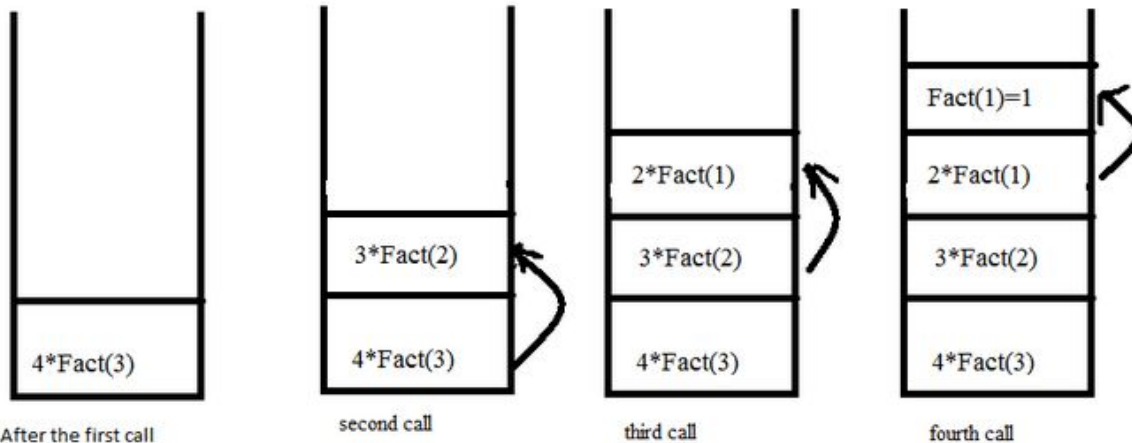
```
1 // factorial recursive call
2 return n * factorial(n - 1);
3
4 // fibonacci base case
5 return fibonacci(n - 1) + fibonacci(n - 2);
6
7 // binary search base case
8 return binarySearch(arr, low, mid - 1, target);
9 // тук има още едно рекурсивно извикване :)
```

Системен стек

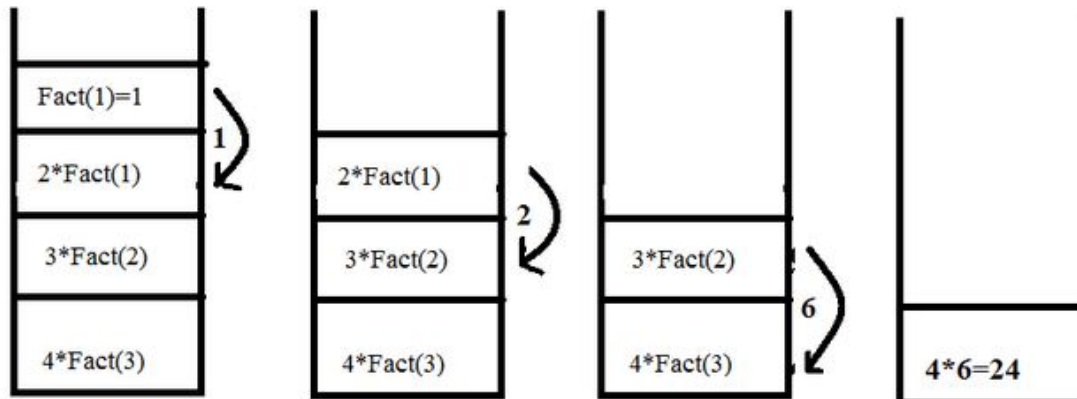
- Управлява локалните променливи и извикванията на функциите
- First In, Last Out (FIFO)
- Всяко рекурсивно извикване се добавя с стека



When function call happens previous variables gets stored in stack



Returning values from base case to caller function



Съвети

- Подсигурете се, че базовият случай винаги се достига
- Оптимизация - част от рекурсивните извиквания могат да бъдат опростени, чрез запомняне на предишни резултати
- Използване на стека
- Четимост

Край