Recursão

Técnicas de Programação

Igor Montagner (igorsm1@insper.edu.br)

Recursão

algoritmo que chama a si mesmo uma ou mais vezes

passando uma instância menor do problema

Igor Montagner (igorsm1@insper.edu.br)

Recursão

- facilita muito a criação de alguns tipos de algoritmos
- técnica avançada difícil de entender
  - simular um algoritmo recursivo não é fácil
  - outras maneiras de pensar sobre corretude

O que aconteceria ao executar esse algoritmo?

```
WTF()
  PRINT("comecou")
  WTF()
  PRINT("fim")
}
```

1. termina pois dá erro
2. termina com sucesso
3. não termina

Igor Montagner (igorsm1@insper.edu.br)

O que aconteceria ao executar esse programa em *C*?

```c
void wtf() {
    printf("comecou\n");
    wtf();
    printf("terminou\n");
}
```

1. termina pois dá erro
2. termina com sucesso
3. não termina

Igor Montagner (igorsm1@insper.edu.br)

O que aconteceria ao executar esse programa em *C*?

```c
void wtf() {
    printf("comecou\n");
    wtf();
    printf("terminou\n");
}
```

1. **termina pois dá erro**
2. termina com sucesso
3. não termina

Não existe como fazer uma sequência infinita de chamadas de função!

Igor Montagner (igorsm1@insper.edu.br)

Simulação de chamadas de função

- a cada chamada de função adicionamos indentação na saída
- a cada chamada abrimos um `{` e colocamos ao lado o nome da função e seus parâmetros
- quando houver um `return` adicionamos um `}` e colocamos ao lado o valor de retorno

Igor Montagner (igorsm1@insper.edu.br)

Exemplo 1

```c
void bar() {
    printf("X\n");
}

void foo() {
    printf("A\n");
    bar();
    printf("B\n");
    bar();
    printf("C\n");
}

int main() {
    printf("1\n");
    foo();
    printf("2\n");
    bar();
    printf("3\n");
    foo();
    printf("4\n");
    return 0;
}
```

Exemplo 2

```
REC(N):

if N = 1 then
    return 1
end

return N * REC(N-1)
```

** Simule para `N=5`

Igor Montagner (igorsm1@insper.edu.br)

Atividade 1: Simulação recursiva

Igor Montagner (igorsm1@insper.edu.br)

E esse programa? Ele está correto?

Uma maneira melhor de pensar

1. Para quais entradas não precisamos fazer chamada recursiva?
2. O programa termina?
3. O programa funciona, supondo que as chamadas recursivas retornam o valor correto?

Atividade 2 - pensando em algoritmos recursivos

Igor Montagner (igorsm1@insper.edu.br)