Estruturas de Dados – Aula 01

Prof. Dr. Eduardo Takeo Ueda
eduardo.ueda@fatec.sp.gov.br

Ementa

Pilhas, filas, alocação dinâmica, recursividade, listas encadeadas, tabelas de espalhamento e árvores.

Processo de Avaliação

Instrumento de avaliação	Período previsto para aplicação	Devolução
1ª Avaliação individual	9ª semana	1 semana depois
2ª Avaliação individual	16ª semana	1 semana depois
3ª Avaliação individual	18ª semana	1 semana depois

Composição da Nota Final

Nota Final(NF) = (A1 + A2) / 2

onde:
A1 = 1ª Avaliação individual
A2 = 2ª Avaliação individual

Observações:
(1) A 3ª Avaliação individual substituirá a menor das notas entre A1 e A2, e abordará todo o conteúdo do semestre
(2) Será aprovado na disciplina o aluno que obtiver Nota Final (NF) maior ou igual a 6 (seis inteiros)

Algoritmos iterativos e recursivos

- Algoritmos iterativos usam estruturas de repetição tais como laços, ou ainda estruturas de dados adicionais tais como pilhas, para resolver problemas.

- Cada algoritmo iterativo possui um algoritmo recursivo equivalente e vice-versa.

Recursividade

- A recursividade é uma estratégia que pode ser utilizada sempre que uma função f pode ser escrita em função dela própria.

Efeito Droste

- Muitas estruturas de dados como listas e árvores tem natureza recursiva, então é importante compreender a recursividade.

Condição de parada da recursividade

- Nenhum programa, nem função, pode ser exclusivamente definido por si própria, pois:
  - O programa seria um loop infinito;
  - A função teria uma definição circular.

- Condição de parada (também chamado de base da recursão)
  - Permite que o procedimento pare de se executar;
  - Por exemplo, f(x) >0 com x decrescente.

Função fatorial

- A função fatorial para um inteiro não negativo pode ser definida de forma recursiva como:

$$fatorial(n) = \begin{cases} 1 & \text{se } n = 0 \\ n \times fatorial(n-1) & \text{se } n > 0 \end{cases}$$

Fatorial recursivo

```
public class FatorialRecursivo {

    public static void main(String[] args){
          System.out.println("Fatorial de "+ 4 + " é " + fatorial(4));
    }

    public static long fatorial(long n) {
          if (n == 0 )  return 1; //condição de parada
          else  return n * fatorial(n - 1); //chamada recursiva
        }
}
```

Fatorial iterativo

```
public class FatorialIterativo {

    public static void main(String[] args) {
        System.out.println("Fatorial de " + 4 + " é " + fatorial(4));
    }

    public static long fatorial(long n) {
        long i, fat;
        fat = 1;
        for ( i = 1; i <= n; i++ )
           fat = fat * i;
        return fat;
    }
}
```

Série de Fibonacci

- Um outro exemplo clássico de recursividade é a famosa série de Fibonacci, definida pela expressão:

$$F(n) = \begin{cases} 0 & \text{se } n = 0 \\ 1 & \text{se } n = 1 \\ F(n-1) + F(n-2) & \text{se } n \geq 2 \end{cases}$$

- 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

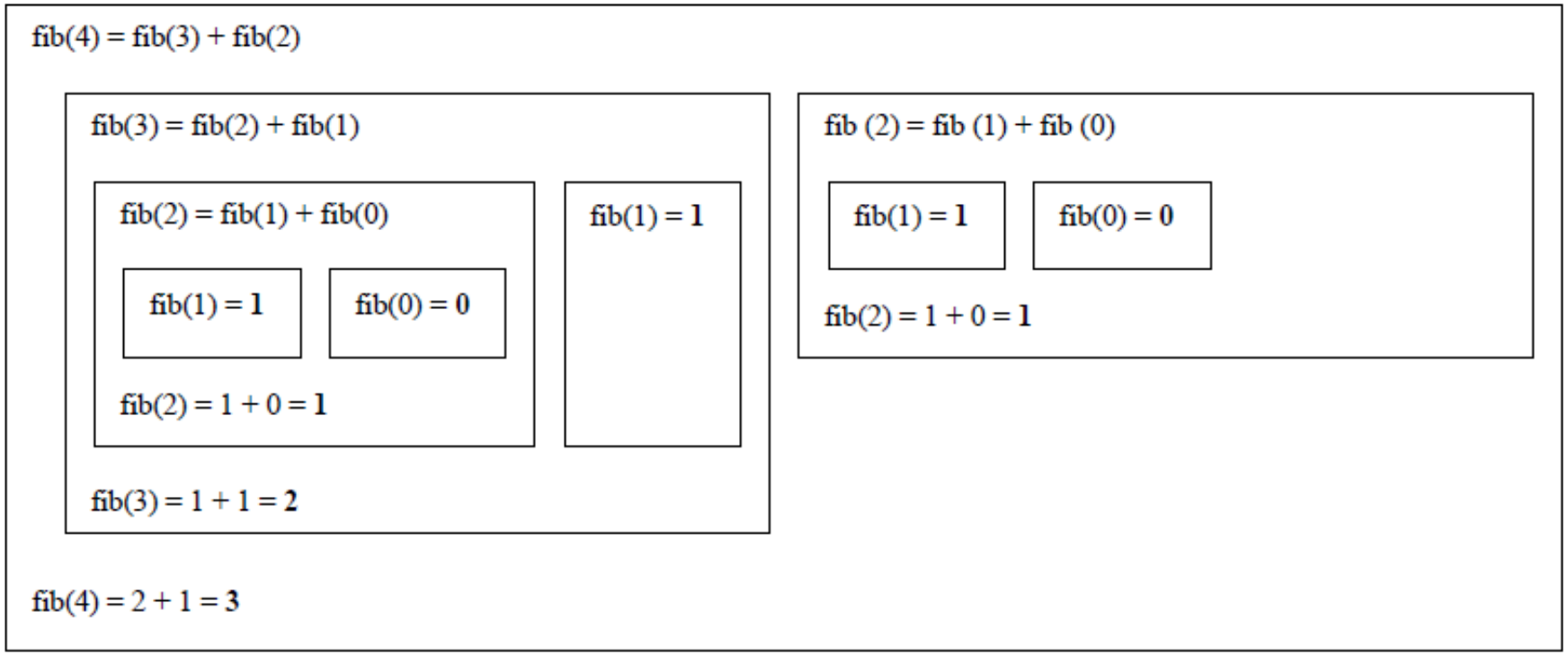
Fibonacci recursivo

```
public class FibonacciRecursivo {

    public static void main(String[] args) {
        System.out.println("Fibonacci de " + 5 + " é " + fibonacci(5));
    }

    public static long fibonacci(long n) {
        if(n == 0) //condição de parada
            return 0;
        else if (n == 1) //condição de parada
            return 1;
        else
            return fibonacci( n - 1 ) +  fibonacci( n - 2 );
    }
}
```

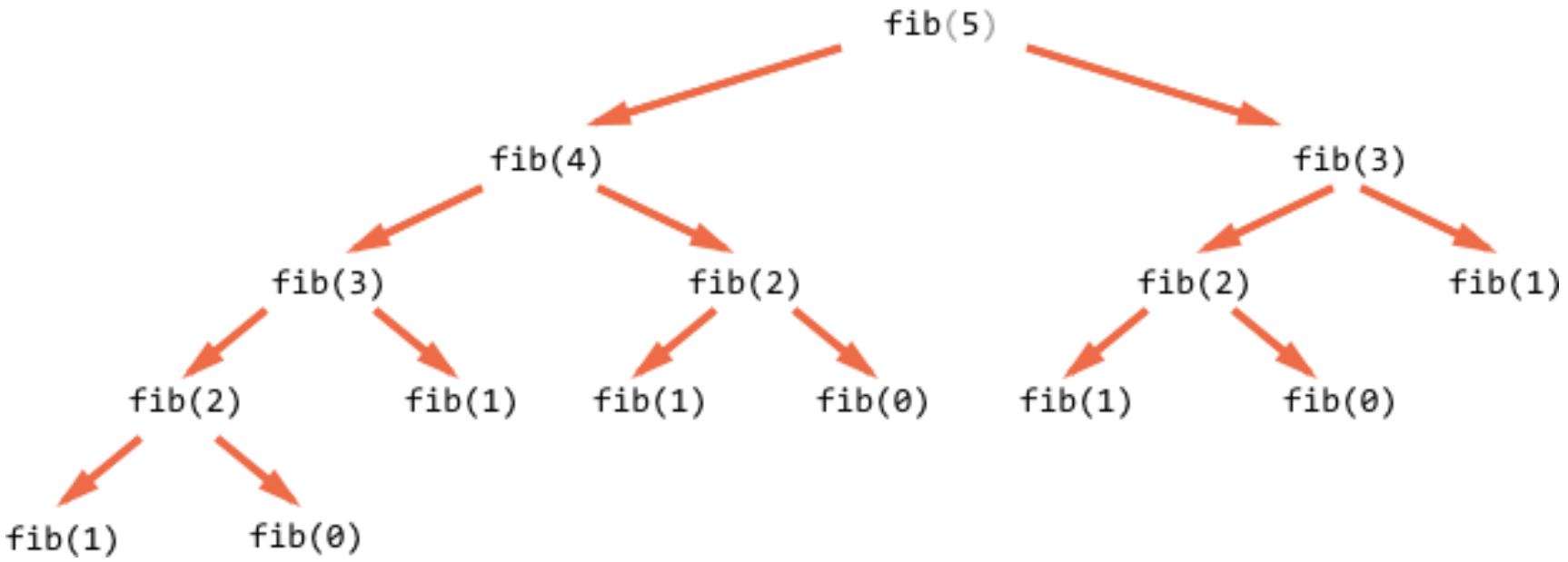
Fibonacci iterativo

```
public class FibonacciIterativo {

    public static void main(String[] args) {
        System.out.println("Fibonacci de " + 5 + " é " + fibonacci(5));
    }

    public static long fibonacci(long n) {
        long i, k, F;
        i = 1;
        F = 0;
        for (k = 1; k <= n; k++) {
            F = F + i;
            i = F - i;
        }
        return F;
    }
}
```

“Desenhando/representando” a recursividade



“Desenhando/representando” a recursividade

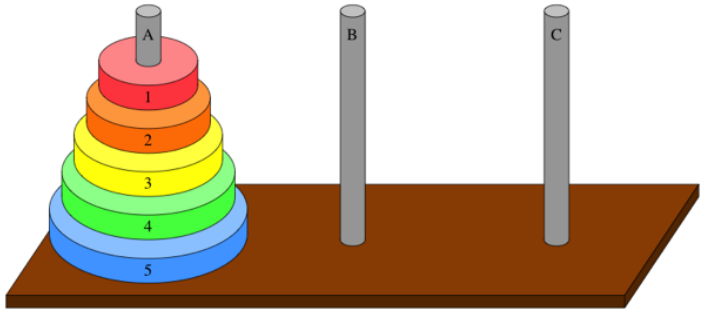


Árvore de recorrência

Atividade

- Implemente um programa recursivo em Java para o jogo das Torres de Hanói.

- Desenhe a árvore de recursão do programa recursivo considerando apenas 4 discos.



- Implemente um programa iterativo em Java para o jogo das Torres de Hanói.

Fim!