Universidade Federal de Mato Grosso do Sul

Estrutura de Dados - Prof. Francisco Eloi.

Turma: Engenharia de Software
RGA: 2021.1906.069-7
Aluno: Maycon Felipe da Silva Mota

---

1. Implemente versões iterativas e recursivas da função fib obtendo também o tempo de execução (note que a versão recursiva já está pronta);

In [ ]:

```
import time

def fibRecursive(n: int): ## Possui complexidade O(2^n)
    if(n <= 1): # 1
        return n # 1
    return (fibRecursive(n-1) + fibRecursive(n-2))

def fibInterative(n: int): ## Possui complexidade O(N)
    f1=1
    f2=1
    tmp=int()
    for i in range(1,int(n)-1):
        tmp = f1+f2
        f1=f2
        f2=tmp
    return f2

for i in [10, 20, 30, 40, 45, 50]:
    print(f"\n Iniciando Fib Recursivo para {i}")
    start_time = time.time()
    vlrFib = fibRecursive(i)
    print(f"Valor do Fib Recursivo: {vlrFib}")
    print("--- %s seconds --- \n " % (time.time() - start_time))

    print(f"\n Iniciando Fib Interativo para {i}")
    start_time = time.time()
    vlrFib = fibInterative(i)
    print(f"Valor do Fib Interativo: {vlrFib}")
    print("--- %s seconds ---" % (time.time() - start_time))
```

2. Execute os dois medindo o tempo de execução paran = 10, 20, 30, 35, 40, 45, 50;

In [ ]:

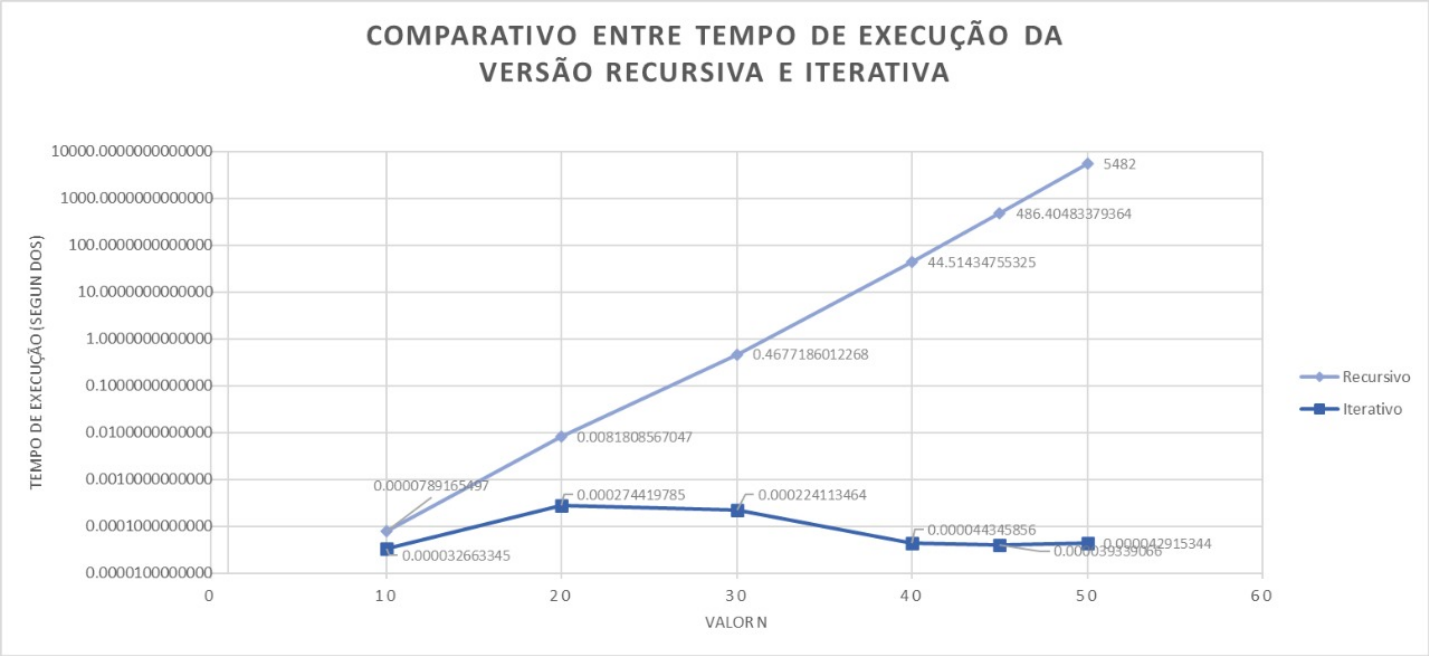
```
import time
```

```python
def fibRecursive(n: int):
    if(n <= 1):
        return n
    return (fibRecursive(n-1) + fibRecursive(n-2))

def fibInterative(n: int):
    f1=1
    f2=1
    tmp=int()
    for i in range(1,int(n)-1):
        tmp = f1+f2
        f1=f2
        f2=tmp
    return f2

for i in [10, 20, 30, 40, 45, 50]:
    print(f"\n Iniciando Fib Recursivo para {i}")
    start_time = time.time()
    vlrFib = fibRecursive(i)
    print(f"Valor do Fib Recursivo: {vlrFib}")
    print("--- %s seconds --- \n " % (time.time() - start_time))

    print(f"\n Iniciando Fib Interativo para {i}")
    start_time = time.time()
    vlrFib = fibInterative(i)
    print(f"Valor do Fib Interativo: {vlrFib}")
    print("--- %s seconds ---" % (time.time() - start_time))
```

```
 Iniciando Fib Recursivo para 10
Valor do Fib Recursivo: 55
--- 7.891654968261719e-05 seconds ---


 Iniciando Fib Interativo para 10
Valor do Fib Interativo: 55
--- 3.266334533691406e-05 seconds ---

 Iniciando Fib Recursivo para 20
Valor do Fib Recursivo: 6765
--- 0.008180856704711914 seconds ---


 Iniciando Fib Interativo para 20
Valor do Fib Interativo: 6765
--- 0.00027441978454589844 seconds ---

 Iniciando Fib Recursivo para 30
Valor do Fib Recursivo: 832040
--- 0.46771860122680664 seconds ---


 Iniciando Fib Interativo para 30
Valor do Fib Interativo: 832040
--- 0.00022411346435546875 seconds ---

 Iniciando Fib Recursivo para 40
Valor do Fib Recursivo: 102334155
--- 44.514347553253174 seconds ---


 Iniciando Fib Interativo para 40
Valor do Fib Interativo: 102334155
--- 4.4345855712890625e-05 seconds ---

 Iniciando Fib Recursivo para 45
Valor do Fib Recursivo: 1134903170
--- 486.40483379364014 seconds ---


 Iniciando Fib Interativo para 45
Valor do Fib Interativo: 1134903170
--- 3.933906555175781e-05 seconds ---
```

```
 Iniciando Fib Recursivo para 50
Valor do Fib Recursivo: 12586269025
--- 5482.83416724205 seconds ---


 Iniciando Fib Interativo para 50
Valor do Fib Interativo: 12586269025
--- 4.291534423828125e-05 seconds ---
```

3. Faça um gráfico para cada um dos algoritmos tendon nas ordenadas (eixo x) e tempo de execução nas abscissas (eixo y);



4. Sem executar estime o tempo gasto aproximado por cada um dos algoritmos quando n = 200;

Sabendo que o algoritmo recursivo possui complexidade O(2^n), poderiamos estimar que o tempo gasto para n = 200, seria 1.60693804e60 ou, 2^200. Enquanto a versão iterativa seria O(n), ou seja, O(200).

5. Faça uma conclusão desse trabalho;

Analisando os algoritmos, conseguimos perceber que algoritmos diferentes conseguem resolver o mesmo problema, entretanto, com uma eficiência diferente.

Apesar de parecer que recursividade seja a melhor solução por apresentar um código mais encurtado, a complexidade do algoritmo muda. Nesse caso, obtivemos O(2^n), ou seja, fazendo com que, a depender da entrada, o tempo de execução cresça de maneira exponencial, diferente da versão interativa. Portanto, a não devemos utilizar a recursividade cegamente, pois ela pode tanto ajudar quanto complicar o tempo de execução.