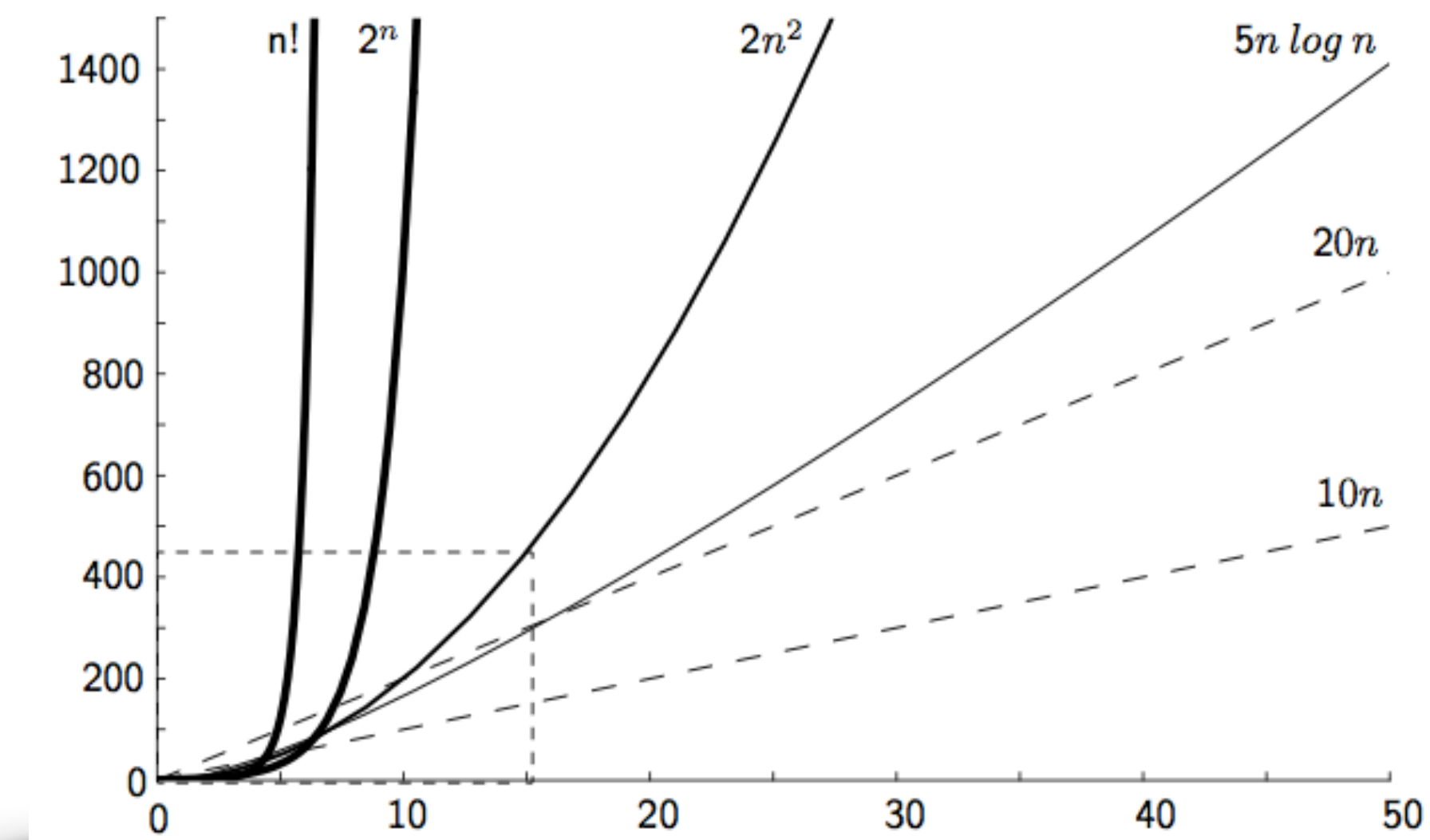


Profa. Dra. Raquel C. de Melo-Minardi
Departamento de Ciência da Computação
Instituto de Ciências Exatas
Universidade Federal de Minas Gerais



MÓDULO 3

COMPLEXIDADE DE ALGORITMOS

Função de complexidade – Parte V

Desafio

1. Esse algoritmo é ótimo?
2. Se não, como ele poderia ser melhorado?
3. Tente implementar melhorias e demonstrar através da nova função de complexidade que ele é mais eficiente

PESQUISA ORDENADA

- ▶ Ordenar a lista para que raramente seja necessário percorrê-la até final para garantir que um registro não esteja presente
- ▶ Exemplo: não é preciso percorrer a lista 2 4 6 8 10 para dizer que o número 3 não está presente
 - ▶ Bastam duas comparações (2 e 4) para saber que o 3 não está presente

Desafio

Tente implementar essa solução de pesquisa em uma lista ordenada.

SOLUÇÃO

```
def pesquisaOrdenada(reg, lista):  
    i = 0  
    while i < len(lista) and reg >= lista[i]:  
        if lista[i] == reg:  
            return i  
        i += 1  
    return -1
```

- ▶ Esse código considera que receberá um arranjo em ordem crescente o que pode ser obtido através do seguinte código

```
lista.sort()
```

SOLUÇÃO

- ▶ A função `pesquisaOrdenada` percorre a lista da esquerda para a direita (menor elemento para o maior) enquanto o elemento corrente for menor que o registro procurado
- ▶ Caso um elemento maior ou igual ao registro procurado seja encontrado no arranjo, o laço é interrompido sem a necessidade de processar até o final
- ▶ Qual é a complexidade desse algoritmo?

ANÁLISE DE COMPLEXIDADE

- ▶ No **melhor caso**, o registro procurado é o primeiro da lista e $f(n) = 1$
- ▶ No **pior caso**, o registro procurado é maior que todos os registros da lista e $f(n) = n$ assim como na versão anterior da função `pesquisa`
 - ▶ Esse é um caso raro
 - ▶ Com a função `pesquisa` TODO registro que não for encontrado, demanda a pesquisa até o final da lista realizando n comparações
- ▶ O **caso médio** também tem o **mesmo custo** que o da função `pesquisa` $f(n) = (n+1) / 2$
- ▶ Então, há ou não vantagem nessa nova função?

ANALISE DE COMPLEXIDADE

- ▶ Embora as funções de custo sejam as mesmas, o desempenho será melhor visto que o pior caso é significativamente menos freqüente e que probabilidade de se interromper a busca antes é maior
- ▶ É preciso considerar o trabalho prévio para se ordenar a lista que, usando algoritmos eficientes, tem um custo adicional de $f(n) = n \log(n)$
- ▶ Os dois algoritmos de pesquisa que vimos até o momento são algoritmos de **pesquisa sequencial**, ou seja, percorrem a lista de forma sequencial do início ao fim de forma bastante ingênua
- ▶ Como poderíamos fazer melhor uso dessa lista ordenada em nossas pesquisas?

COMO FAZER PESQUISA DE FORMA MAIS EFICIENTE?

- ▶ Talvez não tenha chegado a fazer pesquisas em um catálogo telefônico ou mesmo das páginas amarelas
 - ▶ O catálogo é uma lista de todos os assinantes de linhas de telefone de uma cidade (pode ter mais de 1 milhão de pessoas!) e todo o ano recebíamos em nossas casas uma versão atualizada
 - ▶ Como fazíamos a pesquisa nesse livro quando queríamos saber o telefone de alguém?
 - ▶ A pesquisa era feita pelo nome completo e só era possível pois a lista era organizada em ordem alfabética
- ▶ Qual o nosso modelo mental?

Desafio

Pense em como fazer melhor uso dessa lista ordenada em nossas pesquisas?