

Técnicas de Programação

Busca em vetores

Perguntas importantes

1. Como representar um algoritmo sem código?
2. Quantas operações esse algoritmo realiza?
3. Dado um algoritmo incorreto, como mostrar que ele está errado?

Algoritmos de busca

- busca sequencial
- busca binária

Em termos de número de operações, esses algoritmos são equivalentes?

Vamos sempre contar o número de operações **no pior caso**

Definições

Dado um array de tamanho n , seja

- $f(n)$ o número de operações feitas pela busca sequencial no pior caso
- $g(n)$ o número de operações feitas pela busca binária no pior caso

$$f(n) > 0, g(n) > 0 \quad \forall n > 0$$

Definições

Pergunta: Conforme n aumenta, $f(n)$

1. cresce mais rápido
2. de maneira equivalente
3. mais lentamente

que $g(n)$?

Definições II

Vamos usar limites pra isso!

$$\lim_{n \rightarrow +\infty} \frac{f(n)}{g(n)}$$

Qual o valor desse limite se $f(n)$ for _____ que $g(n)$?

1. mais rápido?
2. equivalente?
3. mais lento?

Definições III

1. mais rápido:

$$\lim_{n \rightarrow +\infty} \frac{f(n)}{g(n)} = +\infty$$

2. equivalente

$$0 < \lim_{n \rightarrow +\infty} \frac{f(n)}{g(n)} < +\infty$$

3. mais lento:

$$\lim_{n \rightarrow +\infty} \frac{f(n)}{g(n)} = 0$$

Exercício

1. escrever juntos os algoritmos "oficiais" das buscas sequenciais e binária
2. **Parte 1** do handout de hoje. (10-15 minutos)
3. Discussão na lousa

Notação \mathcal{O}

Dada uma função $g(n)$, dizemos que uma função $f(n)$ é $\mathcal{O}(g(n))$ se e somente se

1. $\lim_{n \rightarrow +\infty} \frac{f(n)}{g(n)} < +\infty$

2. f e g satisfazem $f(n) > 0, g(n) > 0, \quad \forall n > 0$

Classes de algoritmos

- $\mathcal{O}(\log_2 n)$ - dividem a entrada em duas partes e ignoram metade dela a cada iteração
- $\mathcal{O}(n)$ - olham c vezes para cada elemento da entrada, com c constante
- $\mathcal{O}(n^2)$ - loop duplo. Exemplo mais comum é olhar novamente para todo array a cada elemento.

Existem diferenças de tempo de execução entre algoritmos da mesma classe, mas essa diferença não depende do tamanho n da entrada. Para algoritmos de classes diferentes a diferença de tempo **aumenta** de maneira dependente de n .

Exercício

1. **Parte 2** do handout
2. implementações dos algoritmos no PrarieLearn