

COMPARANDO EFICIÊNCIA DE ALGORITMOS

Busca Linear

- $X \in A$: No pior caso, o elemento X está na última posição da lista A . O tempo de execução é $T(n)$.
- $X = A[1]$: O elemento está na primeira posição. O tempo de execução é $T(1)$.
- $X = A[n]$: O elemento está na última posição. O tempo de execução é $T(n)$.
- $X \notin A$: O elemento não está na lista, logo, o algoritmo percorre todos os N elementos. O tempo de execução é $T(n)$.

Busca Linear em Ordem

- $X \in A$: No pior caso, X é o maior elemento na lista A , o que requer percorrer toda a lista. O tempo de execução é $T(n)$.
- $X = A[1]$: O elemento está na primeira posição. O tempo de execução é $T(1)$.
- $X = A[n]$: O elemento está na última posição. O tempo de execução é $T(n)$.
- $X \notin A$: Se X não está na lista, e considerando o pior caso (o valor X é maior que todos os elementos de A), o algoritmo percorre toda a lista. O tempo de execução é $T(n)$.

Busca Binária

- $X \in A$: No pior caso, a busca binária requer $\log_2(n)$ comparações. O tempo de execução é $T(\log n)$.
- $X = A[1]$: Se X está na primeira posição, a busca pode encontrá-lo em $T(\log n)$ tempo, pois ainda divide a lista.
- $X = A[n]$: Similarmente, o tempo de execução é $T(\log n)$.
- $X \notin A$: Mesmo que X não esteja na lista, o algoritmo sempre requer $\log_2(n)$ comparações para concluir a busca. O tempo de execução é $T(\log n)$.

	Busca Linear	Busca Linear em Ordem	Busca Binaria
$X \in A$	$T(n)$	$T(n)$	$T(\log n)$
$X = A[1]$	$T(1)$	$T(1)$	$T(\log n)$
$X = A[N]$	$T(n)$	$T(n)$	$T(\log n)$
$X \notin A$	$T(n)$	$T(n)$	$T(\log n)$