Complexidade - Tempo x Dados

	COMPARAÇÕES/MOVIMENTAÇÕES					
	Melhor	Médio	Pior	Melhor	Médio	Pior
Bublle	O(n2)	O(n2)	O(n2)	O(n2)	O(n2)	O(n2)
Selection	O(n2)	O(n2)	O(n2)	O(n2)	O(n2)	O(n2)
Inserction	O (n)	O(n2)	O(n2)	O (n)	O(n2)	O(n2)
Merge	O (n log n)	O (n log n)	O(n2)	-	-	-

```
Quick-Sort (A , r , p )
      if (r < p )
        indice_pivo = partition (A , r , p)
        Quick-Sort (A , r , indice_pivo)
        Quick-Sort (A , indice_pivo + 1 , p)

Partition (A , r , p)
    indice_pivo = p
    i = r - 1
    j = r
    While (True)
          Repeat j = j - 1
          antil (A [ j ] < A [ indice_pivo] )
          Troca ( indice_pivo , j)
Return j ;
```

```
Shell-Sort (A , n)
    Aux, i , j , h = n/2
    Enquanto h > 0
       I = h
    Enquanto i < n
    Aux= A [ i ]
    J = i
    Enquanto j > h && A [ j - h ] > Aux
          A [ j ] = A [j - h ]
          J = j - h
     A [ j ] = Aux
     i = i + 1
```

BUSCAS

Sequencial - Arranjos numéricos não ordenados

Custo O (n)
Achei = falso
Valor = x
FOR i = 1 até A.comprimento
    Se A [i] = valor
    Achei = True
Se achei -> mostrar valor
Se não -> valor não encontrado

Binária - dividir e conquistar
Sequências ordenadas

Busca-Binaria (A , n , v)
Início = 1
Fim = n
Achei = false
Enquanto início < = fim && não achei
    Meio = (fim - início) /2
    Se v < A [meio]
    Fim = meio - 1
    Senão se V > A [meio]
        Início = meio + 1
    Senão
        Achei = True
    Se achei -> mostra V
    Se não -> V não encontrado

Árvores - conceitos - (árvores T)
Definidas recursivamente
Raiz - primeiro elemento (origem da árvore)
Nó - qualquer elemento
Altura da árvore - quantidade de níveis
Grau da árvore - número máximo de filhos de um nó
Grau de nó - número de filhos de um nó
Filhos - sucessor de um nó
Pai - antecessor de um nó
Folhas - nós finais sem sucessores