

Bubble SORT

n° elemento $\rightarrow n = 5$
n° loops $\rightarrow n-1 = 4$

a

0	1	2	3	4
15	16	6	8	5

↑ ↑ ↑ ↑ ↑

Ascending Order ?

1ª Iteração	15	16	6	8	5
2ª Iteração	15	6	16	8	5
3ª Iteração	15	6	8	16	5
4ª Iteração	15	6	8	5	16

5ª Iteração	15	6	8	5	16
6ª Iteração	6	15	8	5	16
7ª Iteração	6	8	15	5	16
8ª Iteração	6	8	5	15	16
9ª Iteração	6	8	5	15	16

1ª Iteração	6	8	5	15	16
2ª Iteração	6	8	5	15	16
3ª Iteração	6	5	8	15	16
4ª Iteração	6	5	8	15	16
5ª Iteração	6	5	8	15	16

1ª Iteração	6	5	8	15	16
2ª Iteração	5	6	8	15	16
3ª Iteração	5	6	8	15	16
4ª Iteração	5	6	8	15	16
5ª Iteração	5	6	8	15	16

$i = 0$
 $j = 0$
 $\rightarrow O(n^2)$
 $\hookrightarrow (n-1)(n-1)$

for ($i = 0; i < n-1; i++$) {
 flag = 0;
 for ($j = 0; j < n-1-i; j++$) {

if ($a[i] > a[j+1]$) {
 flag = 1;
 temp = $a[j]$
 $a[j] = a[j+1]$
 $a[j+1] = temp$

if (flag == 0) break

INSERTION SORT

n° de elementos = 6

sorted | unsorted

5	4	10	1	6	2
---	---	----	---	---	---

5	4	10	1	6	2
---	---	----	---	---	---

4	5	10	1	6	2
---	---	----	---	---	---

4	5	10	1	6	2
---	---	----	---	---	---

4	5	?	10	6	2
---	---	---	----	---	---

4	?	5	10	6	2
---	---	---	----	---	---

?	4	5	10	6	2
---	---	---	----	---	---

1	4	5	10	6	2
---	---	---	----	---	---

1	4	5	?	10	2
---	---	---	---	----	---

1	4	5	6	10	2
---	---	---	---	----	---

1	4	5	6	?	10
---	---	---	---	---	----

1	4	5	?	6	10
---	---	---	---	---	----

1	4	?	5	6	10
---	---	---	---	---	----

1	?	4	5	6	10
---	---	---	---	---	----

1	2	4	5	6	10
---	---	---	---	---	----

for ($i = 1; i < n; i++$) {
 temp = $a[i]$
 $j = i-1$
 while ($j > 0$ & $a[j] > temp$) {
 $a[j+1] = a[j]$
 $j--$
 }
 $a[j+1] = temp$

$\rightarrow O(n^2)$
 $\rightarrow O(n)$

Selection Sort

0	1	2	3	4	5
7	4	10	8	3	1

Encontrar o elemento mínimo

1	4	10	8	3	7
---	---	----	---	---	---

1	3	10	8	4	7
---	---	----	---	---	---

1	3	4	8	10	7
---	---	---	---	----	---

1	3	4	7	10	8
---	---	---	---	----	---

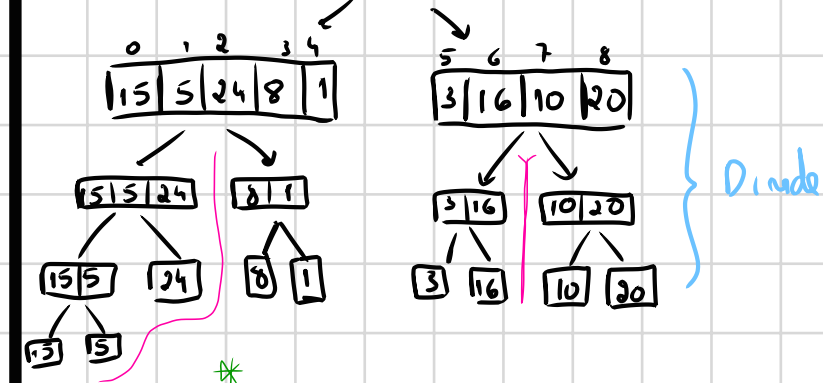
1	3	4	7	8	10
---	---	---	---	---	----

for ($i = 0; i < n-1; i++$) {
 int min = i;
 for ($j = i+1; j < n-1; j++$) {
 if ($a[j] < a[min]$) {
 min = j;
 }
 }
 if (min != i) {
 swap ($a[i], a[min]$);
 }
}

$\rightarrow O(n^2)$

Merge Sort

0	1	2	3	4	5	6	7	8
15	5	24	8	1	3	16	10	20



MergeSort (A, lb, ub) {
 if ($lb < ub$) {
 mid = $(lb + ub) / 2$
 mergeSort (A, lb, mid);
 mergeSort ($A, mid+1, ub$);
 merge (A, lb, mid, ub);
 }

merge (A, lb, mid, ub) {
 $i = lb$
 $j = mid + 1$
 $k = lb$
 while ($i <= mid$ & $j <= ub$) {
 if ($a[i] <= a[j]$) {
 $b[k] = a[i]$
 $i++$
 } else {
 $b[k] = a[j]$
 $j++$
 }
 $k++$
 }
 for ($k = lb; k <= ub; k++$) {
 $a[k] = b[k]$
 }

