Searching in Primary Memory

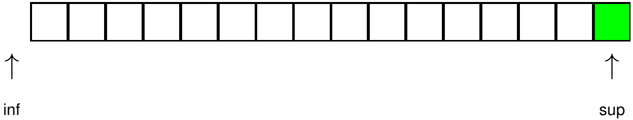
Letícia Rodrigues Bueno

Federal University of ABC (UFABC)

Sequential Search

```
SequentialSearch($key$, $V$)
$i \leftarrow 1$;
while ($i \leq n$) and ($V[i] \neq key$) do
    $i \leftarrow i + 1$;
if $i > n$ then “Success!”
        else “Element not found!”
```

```
BinarySearch($key, list[0 \dots n]$)
$inf \leftarrow -1$
$sup \leftarrow n$
while $inf < sup - 1$ do
    $middle \leftarrow \lfloor \frac{inf+sup}{2} \rfloor$
    if $key \leq list[middle]$ then
        $sup \leftarrow middle$
    else
        $inf \leftarrow middle$
if $key = list[sup]$ then
    return $list[sup]$
else
    return "Element not found!"
```

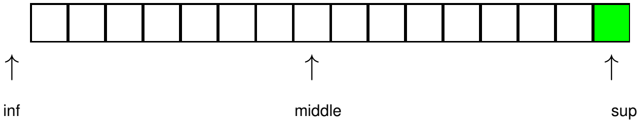


inf

sup

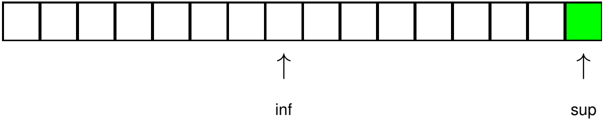
initialization

```
BinarySearch($key, list[0 \ldots n]$)
$inf \leftarrow -1$
$sup \leftarrow n$
while $inf < sup - 1$ do
    $middle \leftarrow \lfloor \frac{inf+sup}{2} \rfloor$
    if $key \leq list[middle]$ then
        $sup \leftarrow middle$
    else
        $inf \leftarrow middle$
if $key = list[sup]$ then
    return $list[sup]$
else
    return "Element not found!"
```



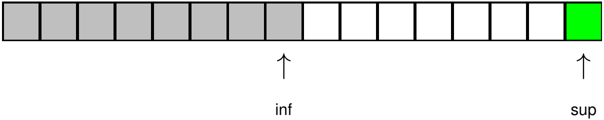
1st
iter.

```
BinarySearch($key, list[0 \dots n]$)
$inf \leftarrow -1$
$sup \leftarrow n$
while $inf < sup - 1$ do
    $middle \leftarrow \lfloor \frac{inf+sup}{2} \rfloor$
    if $key \leq list[middle]$ then
        $sup \leftarrow middle$
    else
        $inf \leftarrow middle$
if $key = list[sup]$ then
    return $list[sup]$
else
    return "Element not found!"
```



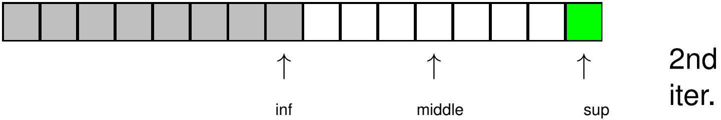
1st iter.

```
BinarySearch($key$, $list[0 \dots n]$)
$inf \leftarrow -1$
$sup \leftarrow n$
while $inf < sup - 1$ do
    $middle \leftarrow \lfloor \frac{inf+sup}{2} \rfloor$
    if $key \leq list[middle]$ then
        $sup \leftarrow middle$
    else
        $inf \leftarrow middle$
if $key = list[sup]$ then
    return $list[sup]$
else
    return "Element not found!"
```

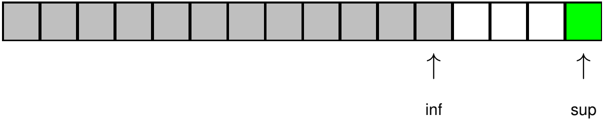


2nd iter.

```
BinarySearch($key, list[0 \ldots n]$)
$inf \leftarrow -1$
$sup \leftarrow n$
while $inf < sup - 1$ do
    $middle \leftarrow \lfloor \frac{inf+sup}{2} \rfloor$
    if $key \leq list[middle]$ then
        $sup \leftarrow middle$
    else
        $inf \leftarrow middle$
if $key = list[sup]$ then
    return $list[sup]$
else
    return “Element not found!”
```

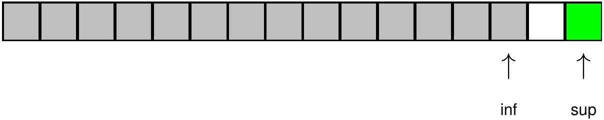


```
BinarySearch($key, list[0 \dots n]$)
$inf \leftarrow -1$
$sup \leftarrow n$
while $inf < sup - 1$ do
    $middle \leftarrow \lfloor \frac{inf+sup}{2} \rfloor$
    if $key \leq list[middle]$ then
        $sup \leftarrow middle$
    else
        $inf \leftarrow middle$
if $key = list[sup]$ then
    return $list[sup]$
else
    return “Element not found!”
```



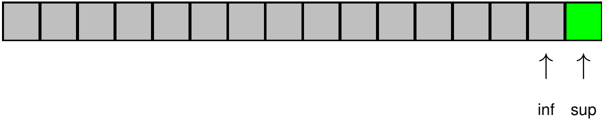
3rd iter.

```
BinarySearch($key, list[0 \dots n]$)
$inf \leftarrow -1$
$sup \leftarrow n$
while $inf < sup - 1$ do
    $middle \leftarrow \lfloor \frac{inf+sup}{2} \rfloor$
    if $key \leq list[middle]$ then
        $sup \leftarrow middle$
    else
        $inf \leftarrow middle$
if $key = list[sup]$ then
    return $list[sup]$
else
    return "Element not found!"
```



4th iter.

```
BinarySearch($key, list[0 \ldots n]$)
$inf \leftarrow -1$
$sup \leftarrow n$
while $inf < sup - 1$ do
    $middle \leftarrow \lfloor \frac{inf+sup}{2} \rfloor$
    if $key \leq list[middle]$ then
        $sup \leftarrow middle$
    else
        $inf \leftarrow middle$
if $key = list[sup]$ then
    return $list[sup]$
else
    return "Element not found!"
```



5th iter.

Bibliography

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L. and STEIN, C. *Introduction to Algorithms*, 3rd edition, MIT Press, 2009.

SZWARCFITER, J. L. and MARKENZON, L. Estruturas de Dados e seus Algoritmos, LTC, 1994. (in Portuguese)

ZIVIANI, N. Projeto de Algoritmos: com implementações em Pascal e C, 2nd edition, Cengage Learning, 2009. (in Portuguese)