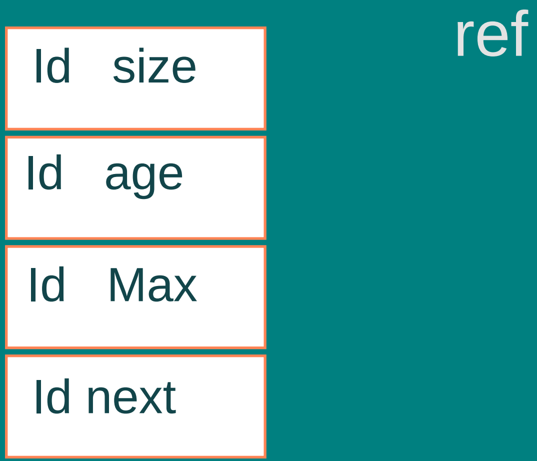
Symbol Table

ref := EnterId(Stacktop)



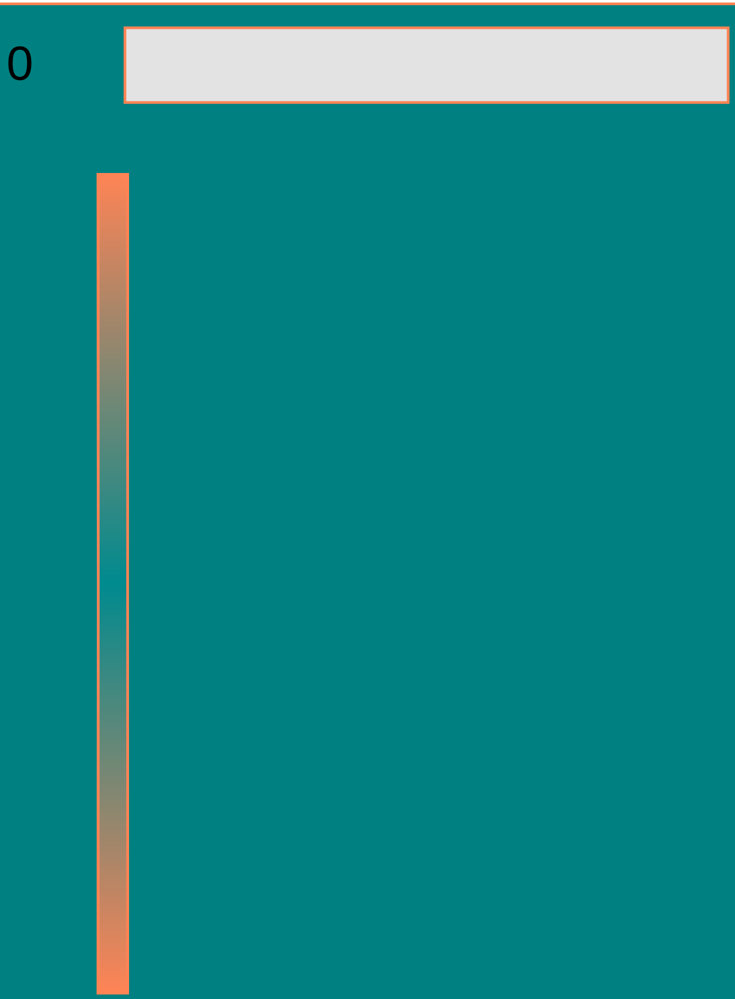
ref

Id size

Id age

Id Max

Id next



Symbol Table

ref := EnterId(Stacktop)

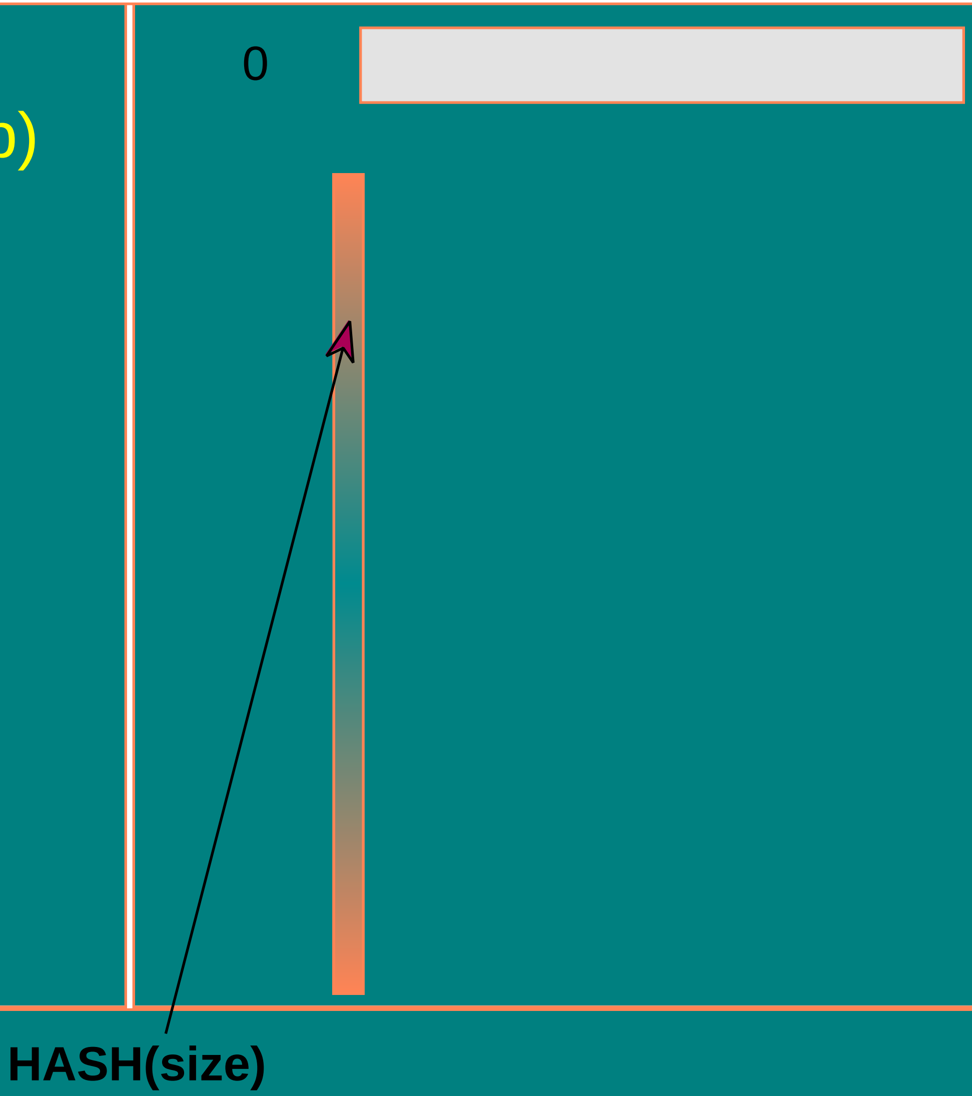
Id size

Id age

Id Max

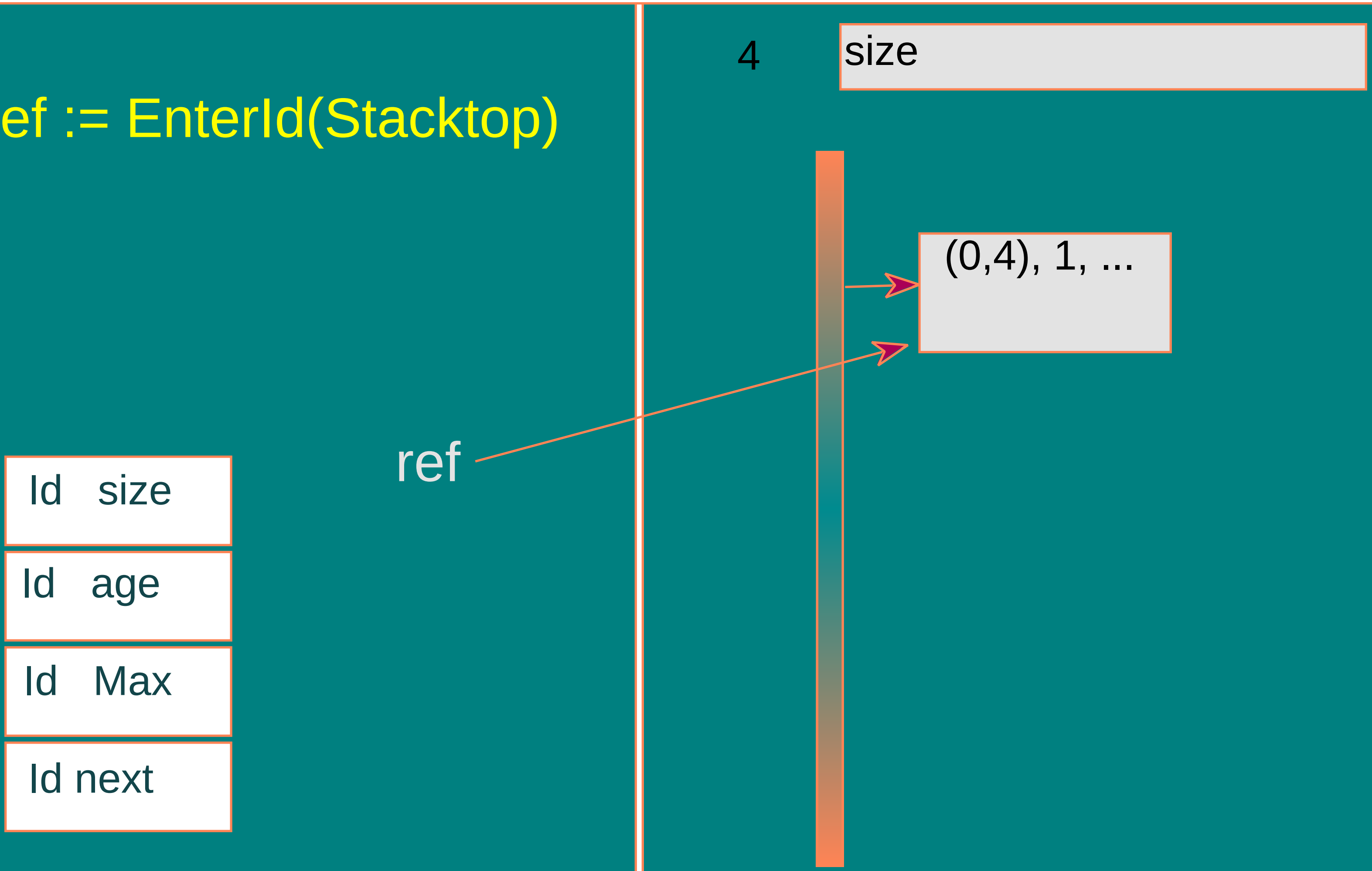
Id next

ref



Symbol Table

ref := EnterId(Stacktop)



4

size

(0,4), 1, ...

ref

Id size

Id age

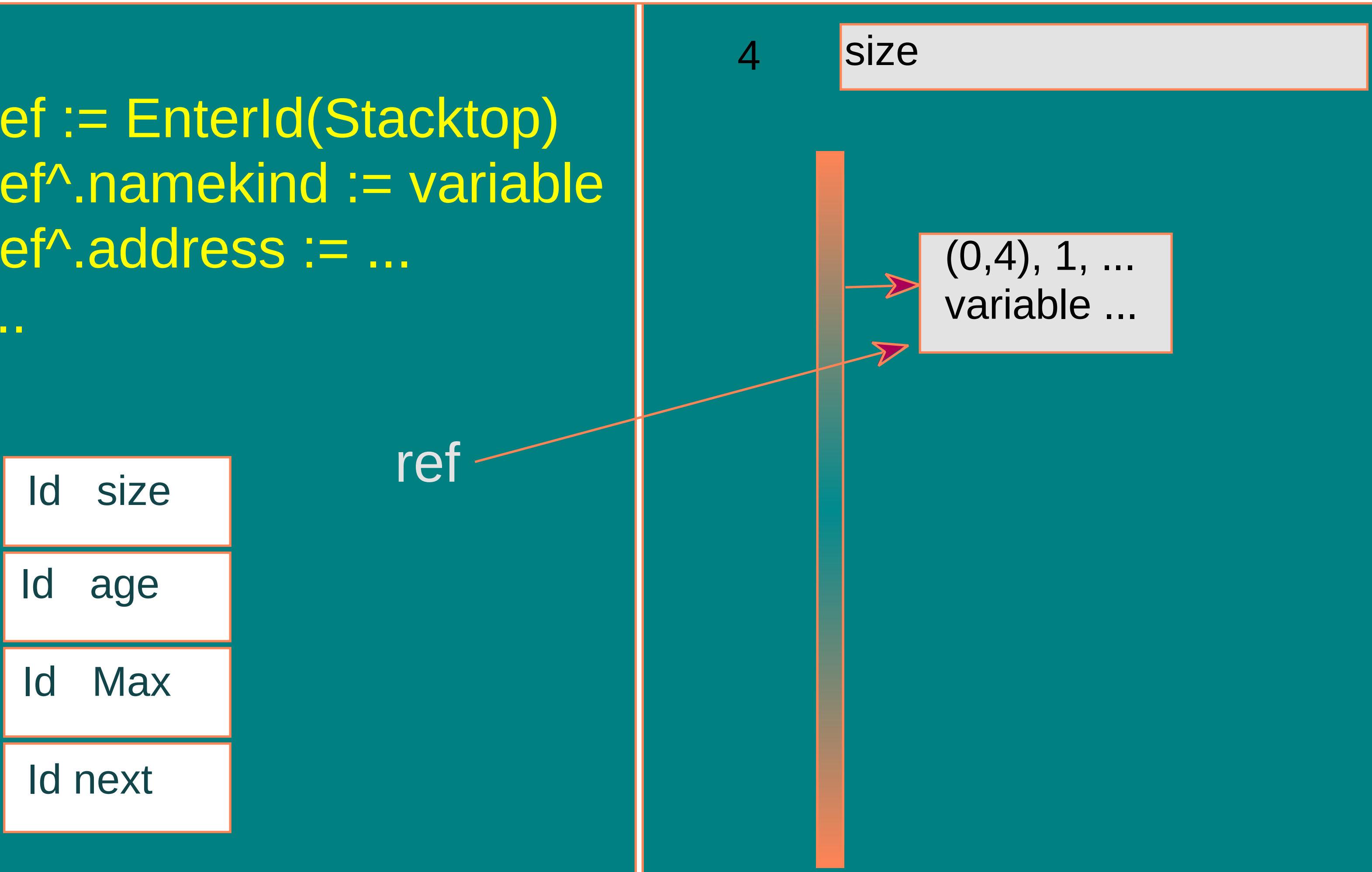
Id Max

Id next

HASH(size)

Symbol Table

```
ref := EnterId(Stacktop)
ref^.namekind := variable
ref^.address := ...
...
```



HASH(size)

Symbol Table

ref := EnterId(Stacktop-1)

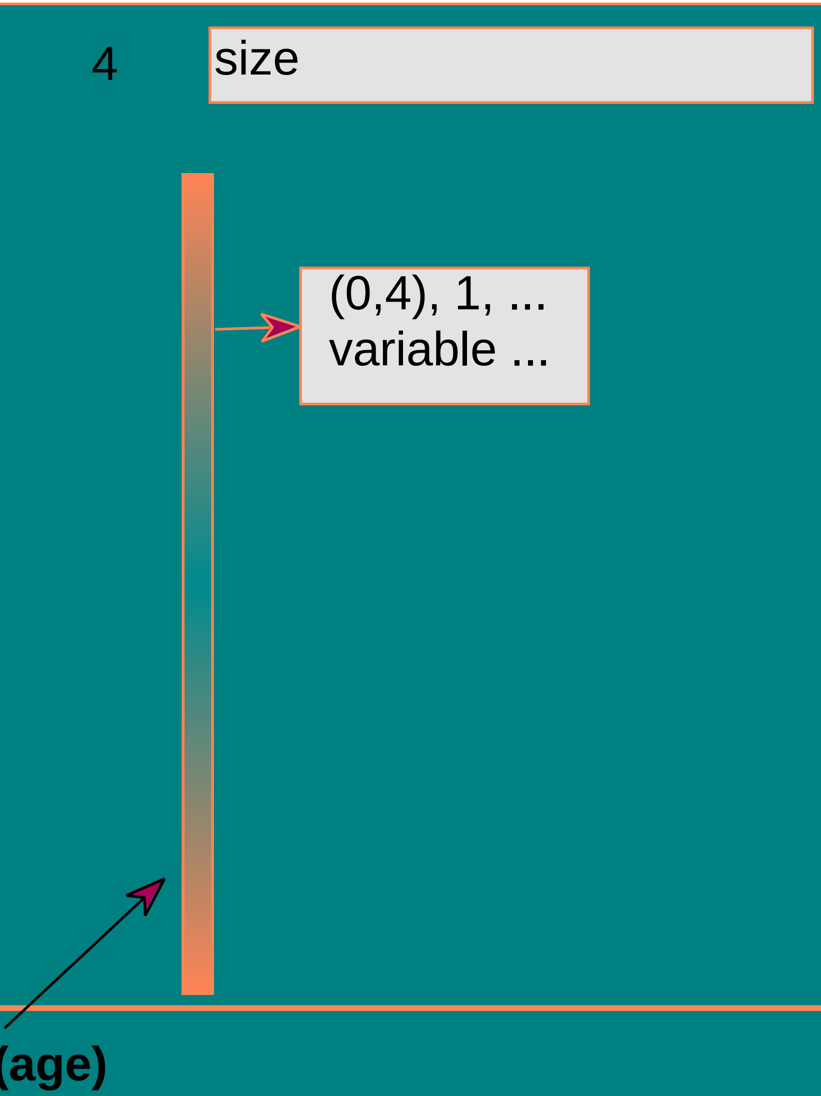
Id size

Id age

Id Max

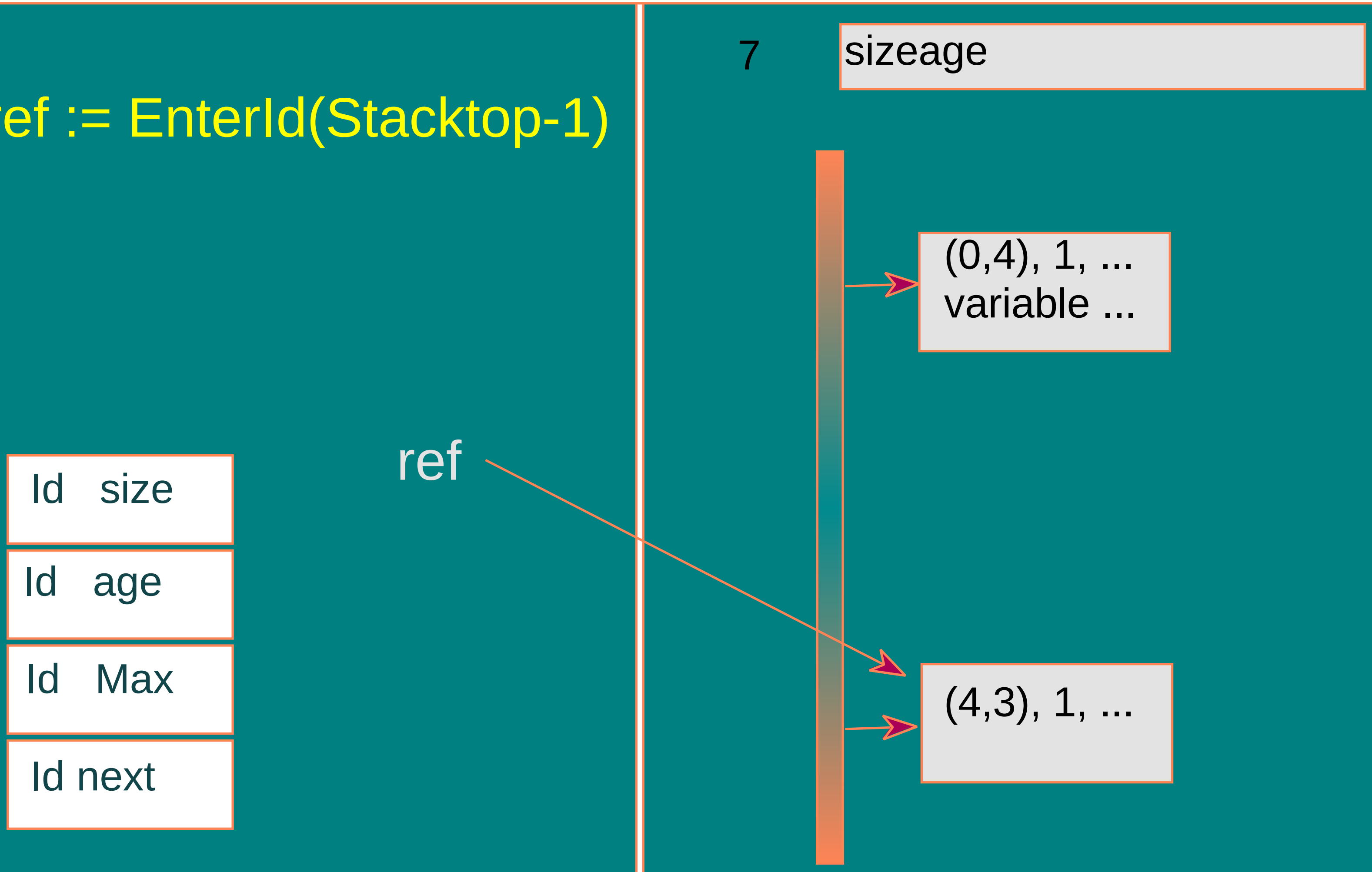
Id next

ref



Symbol Table

ref := EnterId(Stacktop-1)



7 sizeage

(0,4), 1, ...
variable ...

ref

(4,3), 1, ...

Id size

Id age

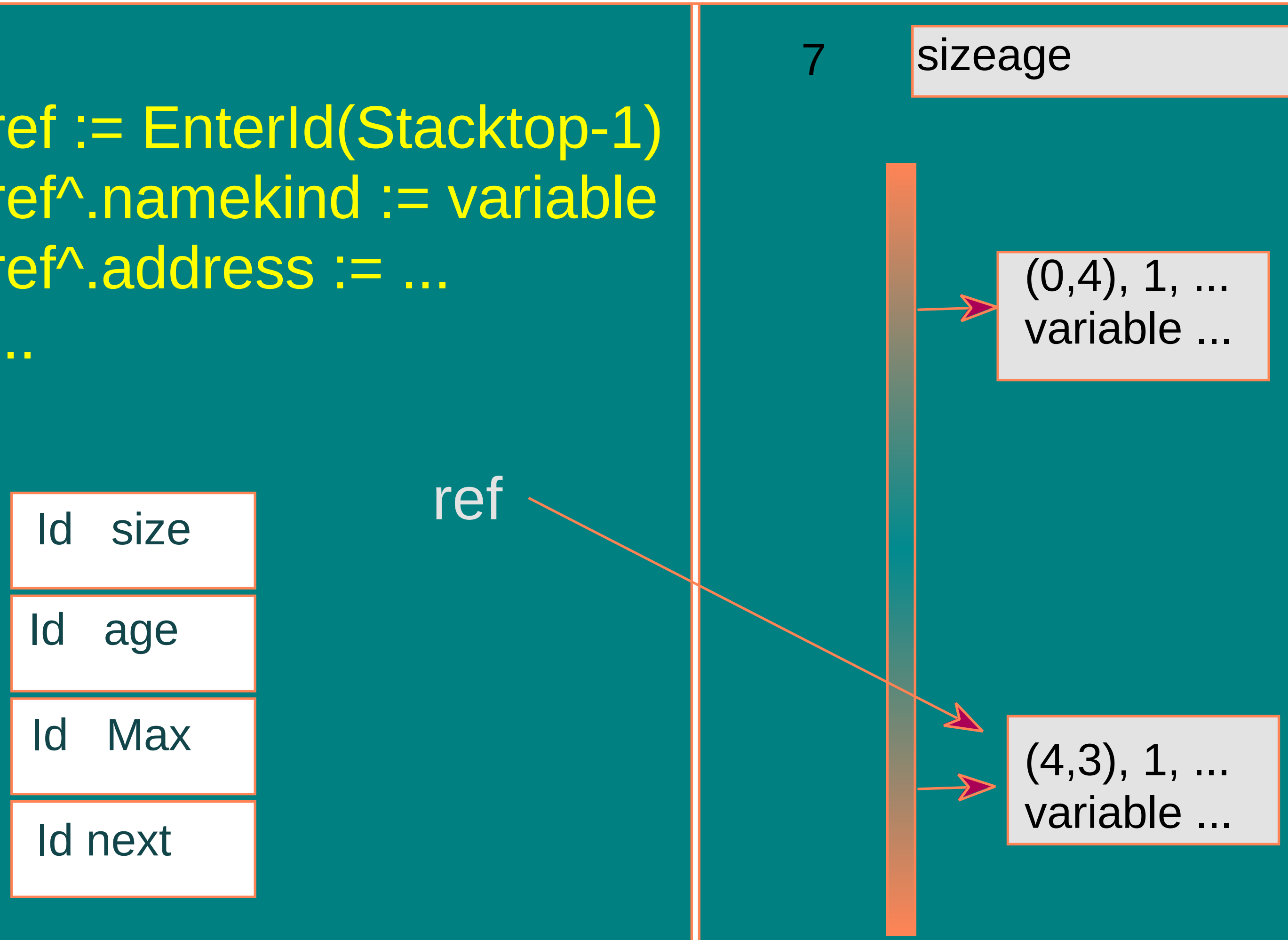
Id Max

Id next

HASH(age)

Symbol Table

```
ref := EnterId(Stacktop-1)
ref^.namekind := variable
ref^.address := ...
...
```



7 sizeage

(0,4), 1, ...
variable ...

ref

Id size

Id age

Id Max

Id next

(4,3), 1, ...
variable ...

HASH(age)

Symbol Table

ref := EnterId(Stacktop-2)

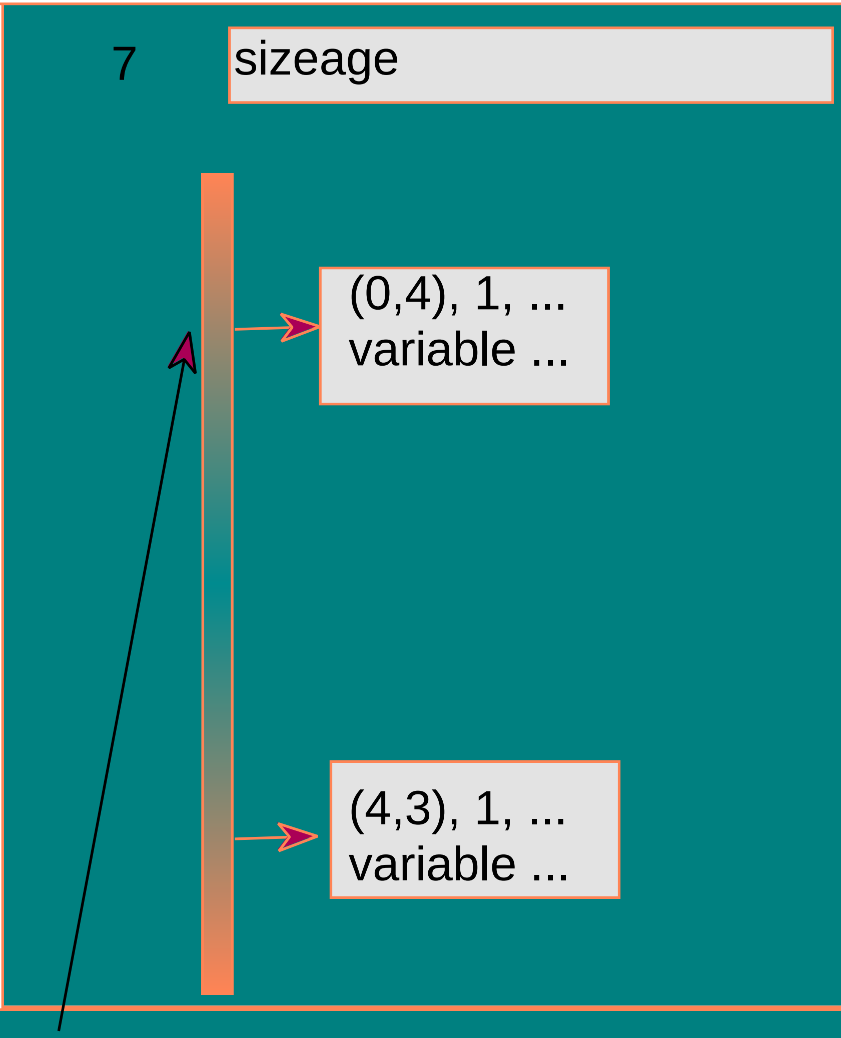
Id size

Id age

Id Max

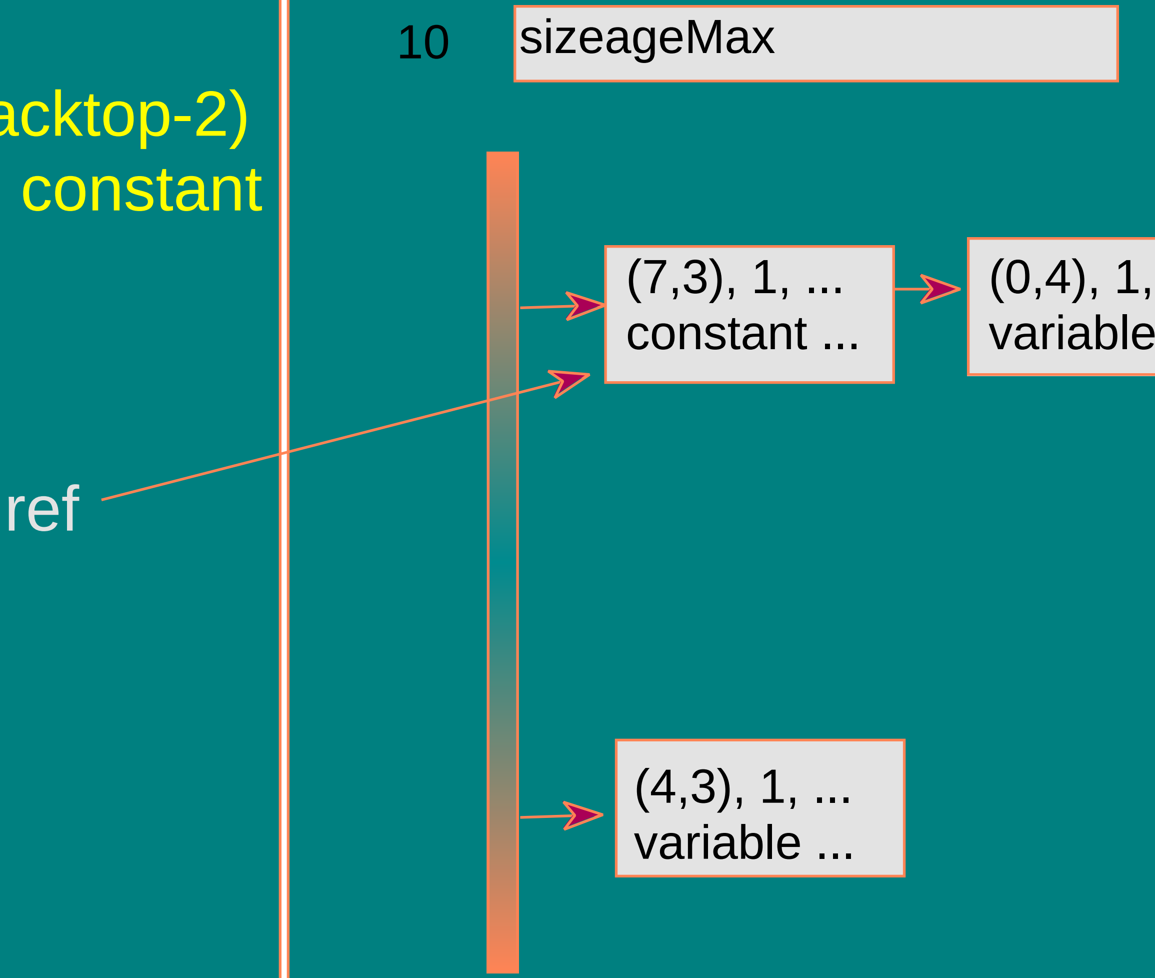
Id next

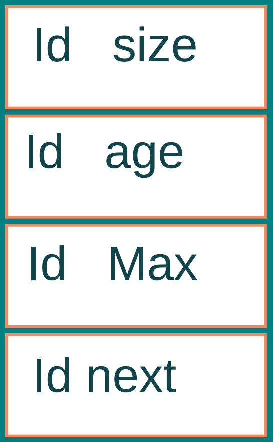
ref



Symbol Table

```
ref := EnterId(Stacktop-2)
ref^.namekind := constant
ref^.value := ...
```





HASH(Max)

EnterId Notes:
Later entries are ahead of earlier entries.
The symbol table is “Stack-like”
EnterId is like push. Later removals will be pops.

EnterId fills in some fields. The caller fills in rest.
ALL other fixed fields & All fields for this variant.

The caller **ALWAYS** fills in the tag field, namekind.

The name itself is stored indirectly in a “spelling table.”

Symbol Table

ref := LookupID(stacktop)

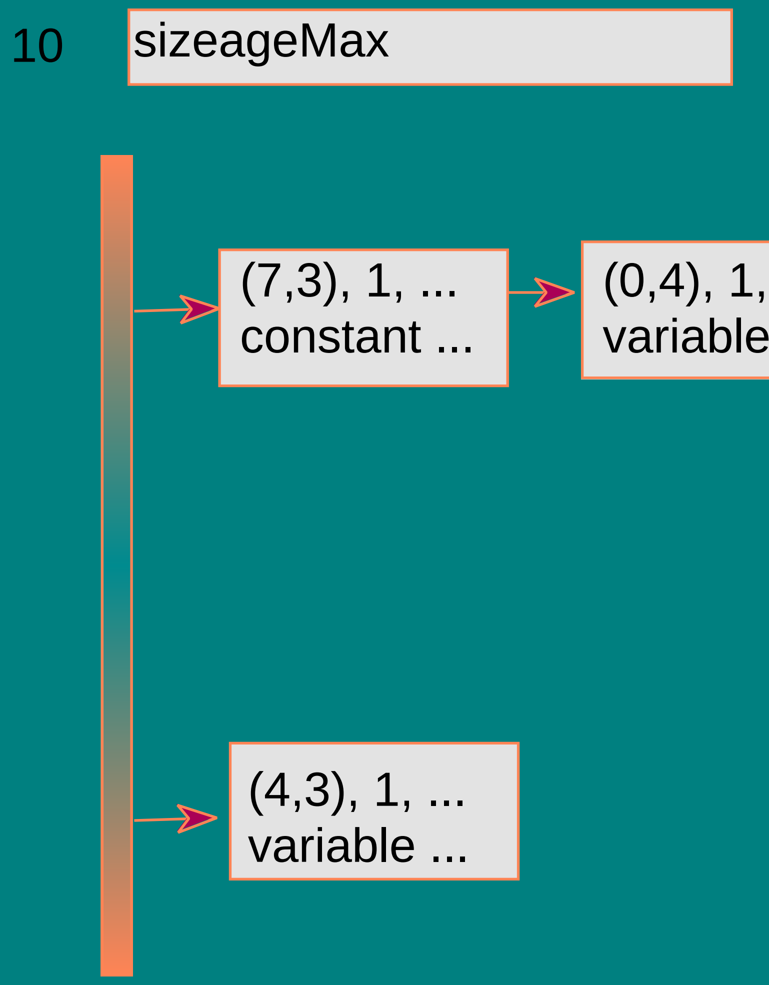
Id size

Id next

Id Max

Id age

ref



10

sizeageMax

(7,3), 1, ...
constant ...

(0,4), 1,
variable

(4,3), 1, ...
variable ...

Symbol Table

ref := LookupID(stacktop)

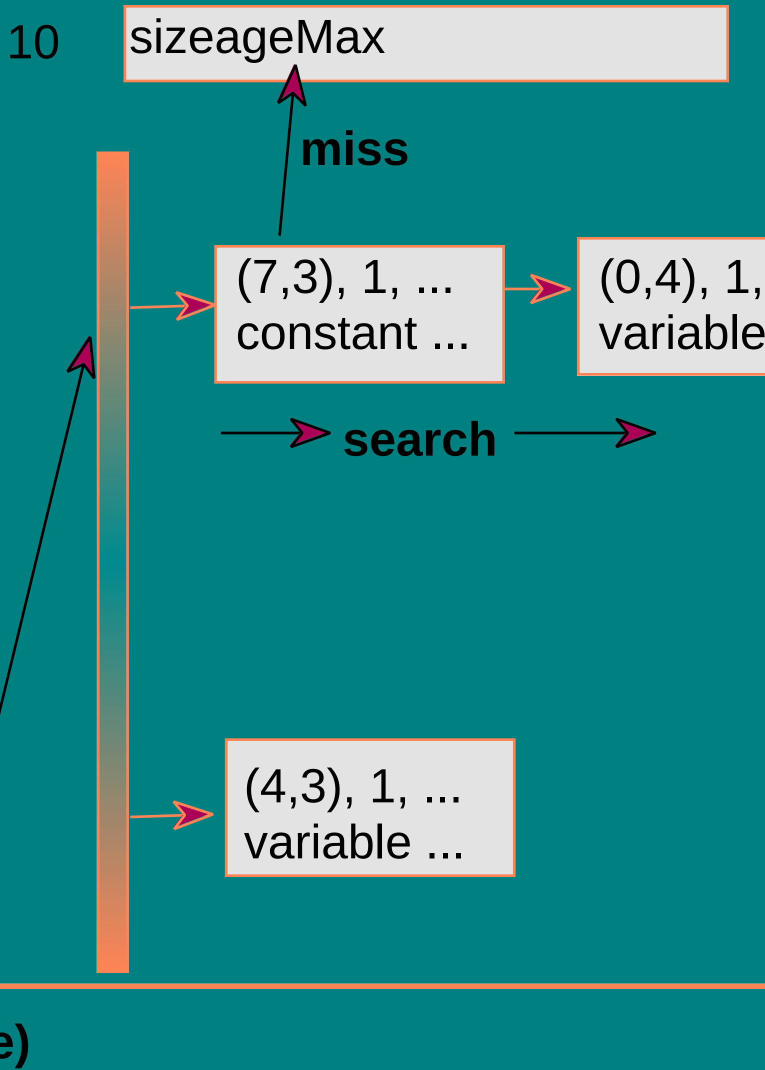
Id size

Id next

Id Max

Id age

ref



10 sizeageMax

miss

(7,3), 1, ...
constant ...

(0,4), 1,
variable

search

(4,3), 1, ...
variable ...

HASH(size)

Symbol Table

ref := LookupID(stacktop)

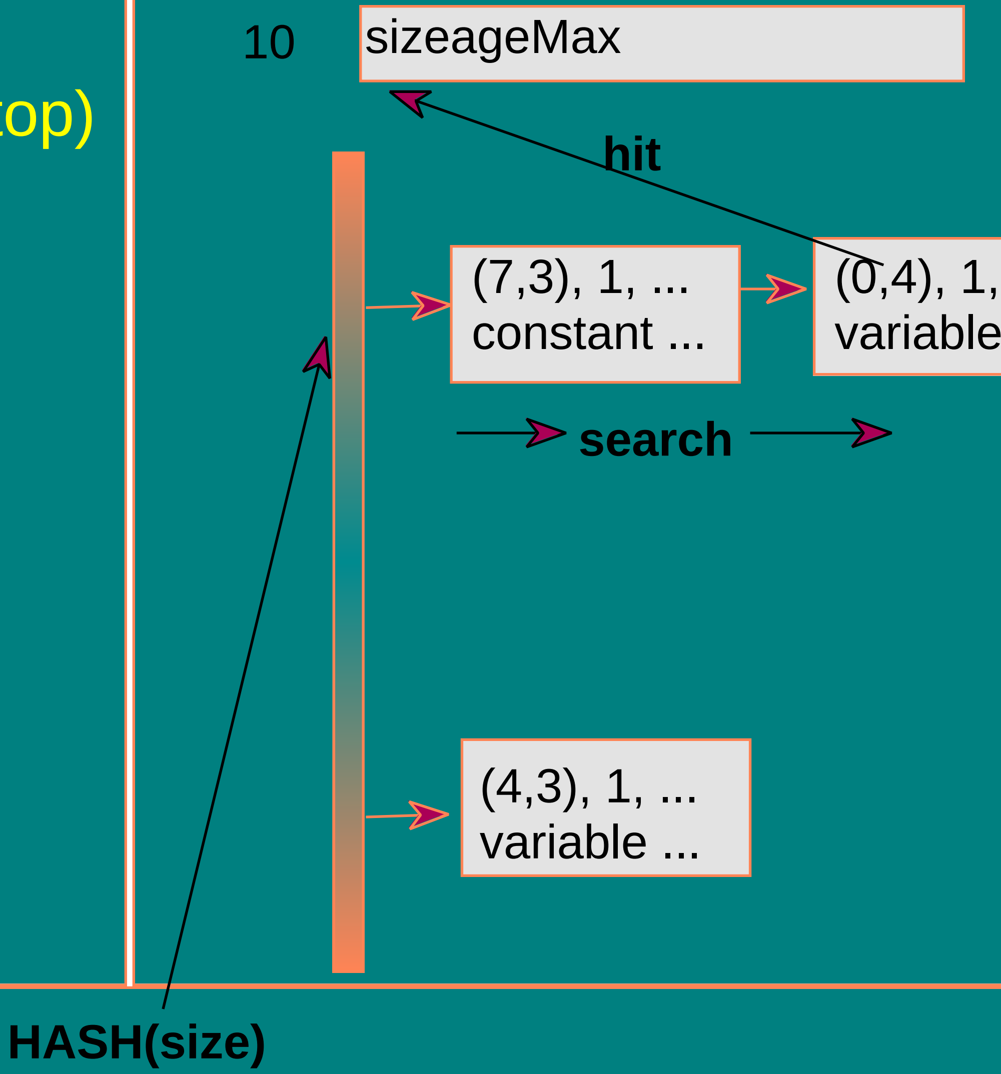
Id size

Id next

Id Max

Id age

ref



10

sizeageMax

hit

(7,3), 1, ...
constant ...

(0,4), 1,
variable

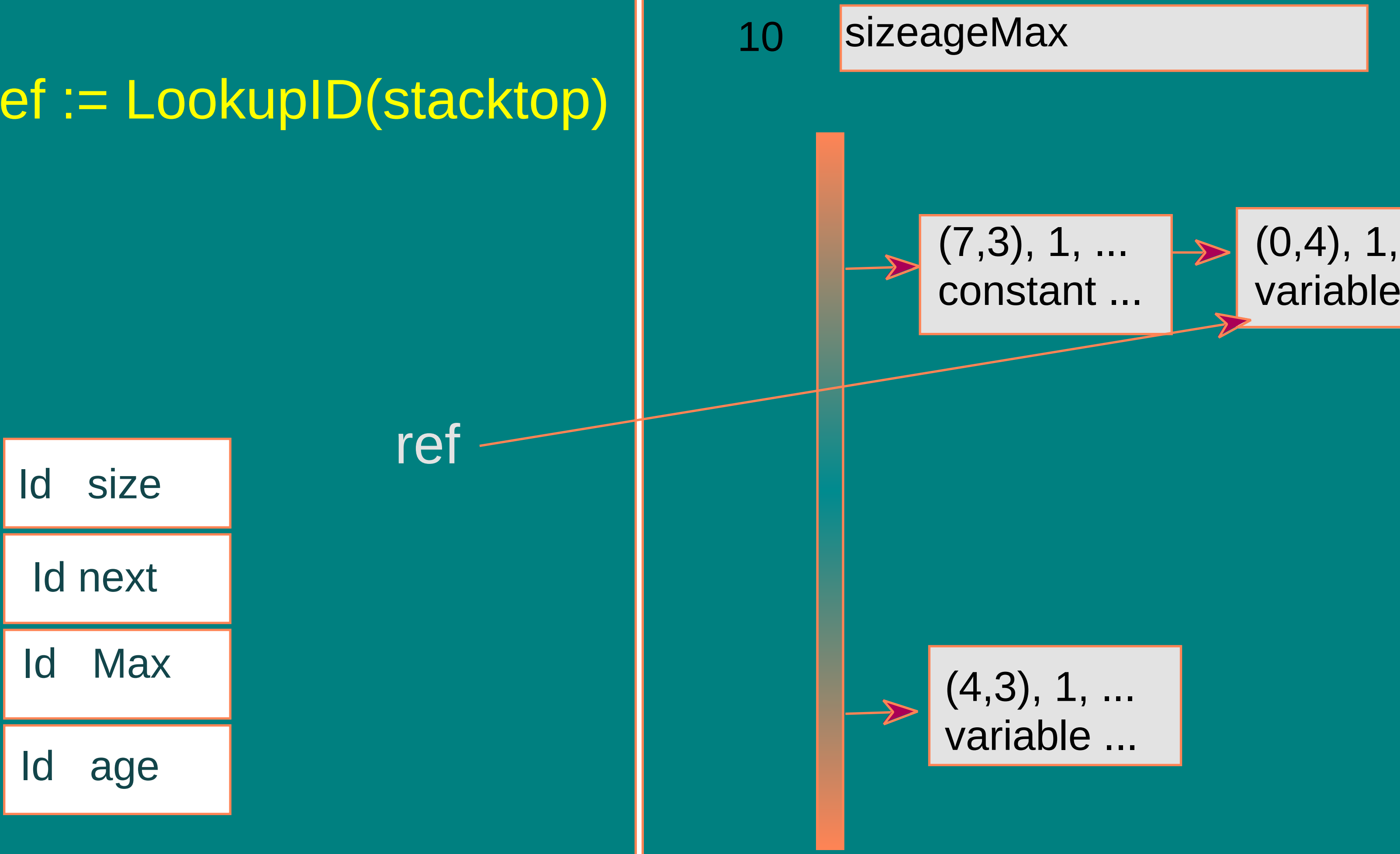
search

(4,3), 1, ...
variable ...

HASH(size)

Symbol Table

ref := LookupID(stacktop)



10 sizeageMax

ref

Id size

Id next

Id Max

Id age

(7,3), 1, ...
constant ...

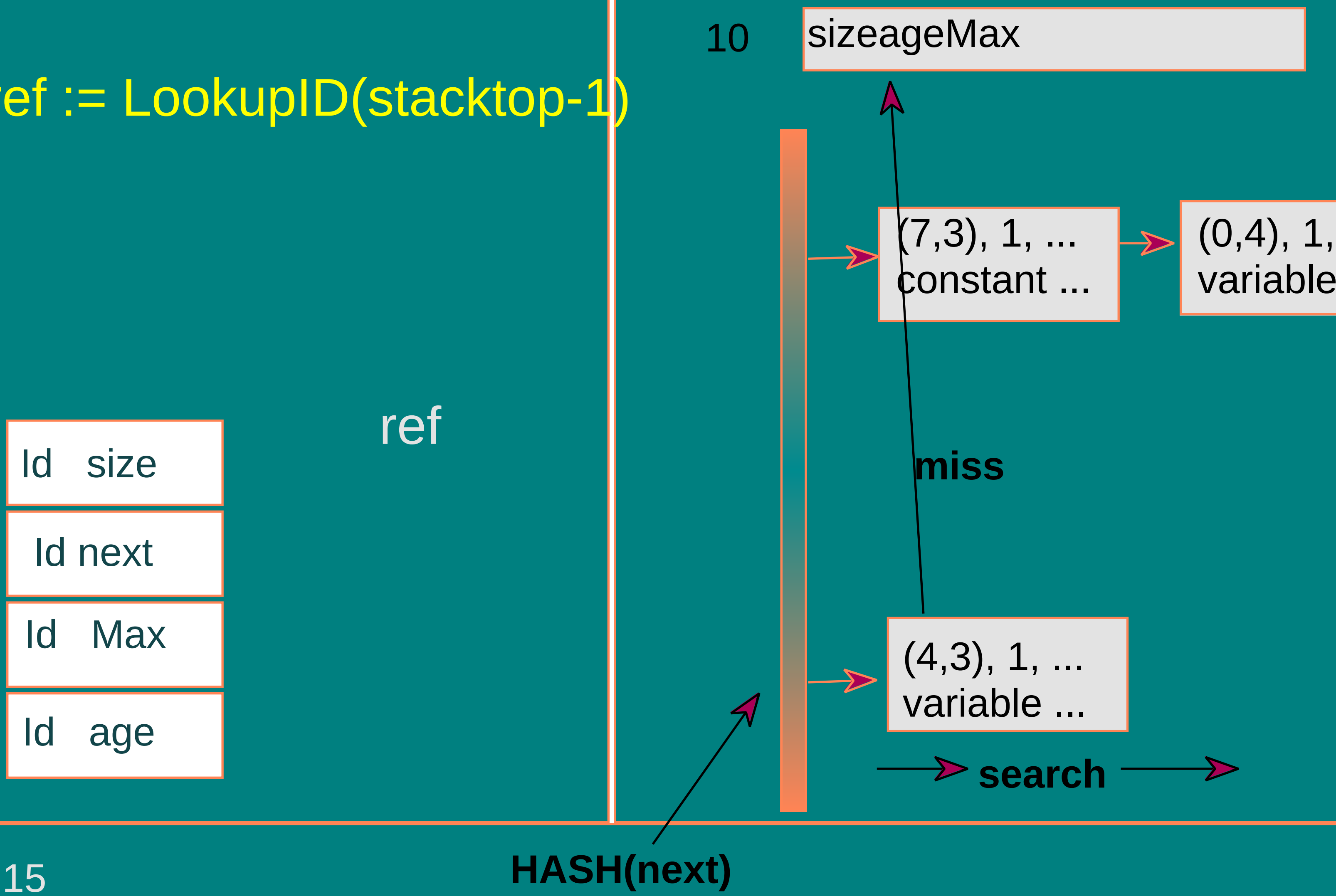
(0,4), 1,
variable

(4,3), 1, ...
variable ...

HASH(size)

Symbol Table

ref := LookupID(stacktop-1)



ref

Id size
Id next
Id Max
Id age

10 sizeageMax

(7,3), 1, ...
constant ...

(0,4), 1,
variable

miss

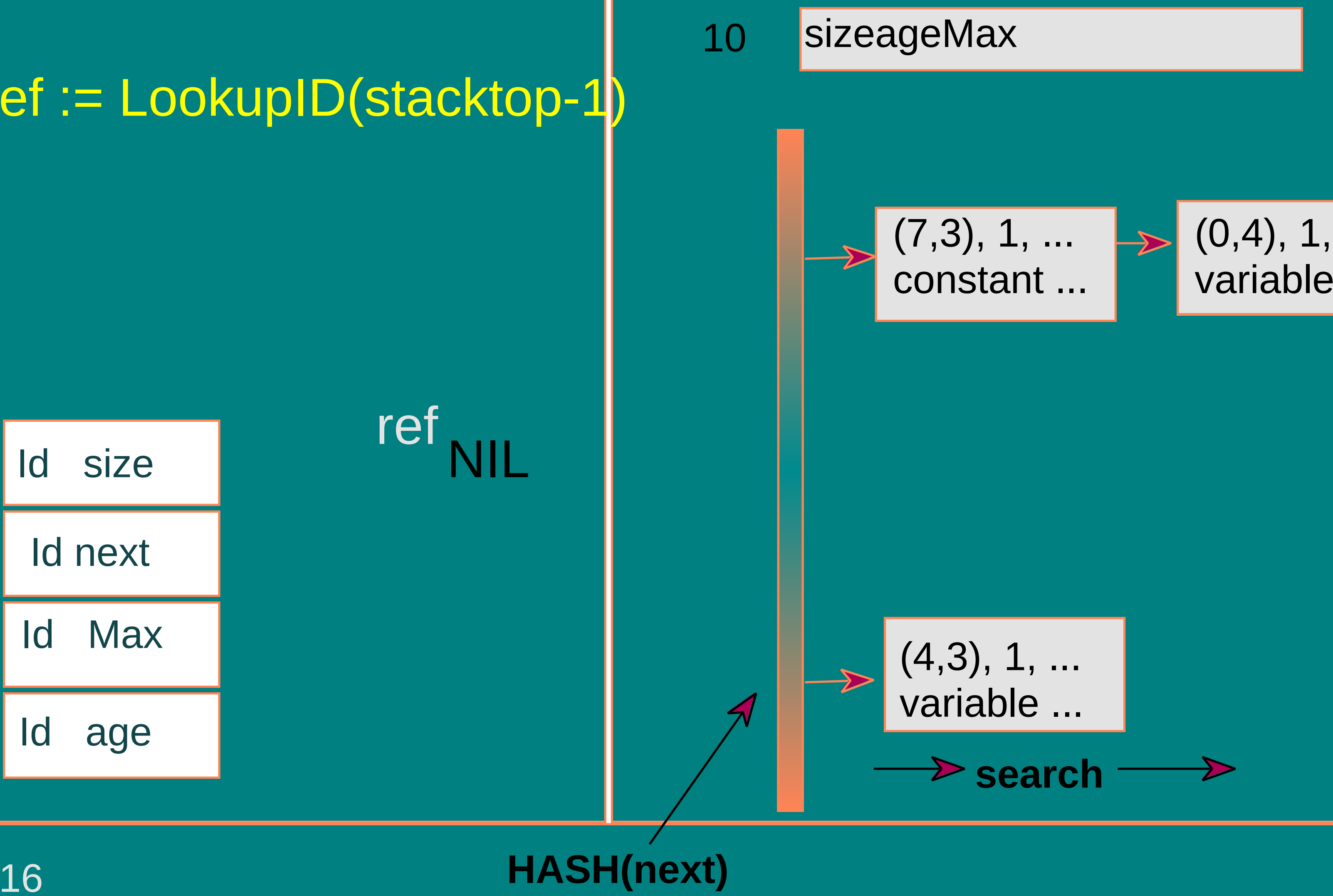
(4,3), 1, ...
variable ...

search

HASH(next)

Symbol Table

ref := LookupID(stacktop-1)



LookupId Notes:

The caller gets back a pointer if the name is stored. Otherwise NIL.

The pointer gives access to all stored data.