Data Structures

# Stack – Array Implementation

Team Emertxe

# Stack – Array Implementation

Data Structure –Stack-Array Implementation

# Operations

- Create Stack
- Insert an Element

# Stack –push(stack,element)

# push(stack,element)

**Input Specification:**

stack : Pointer that contains address of structure variable (stack_t)

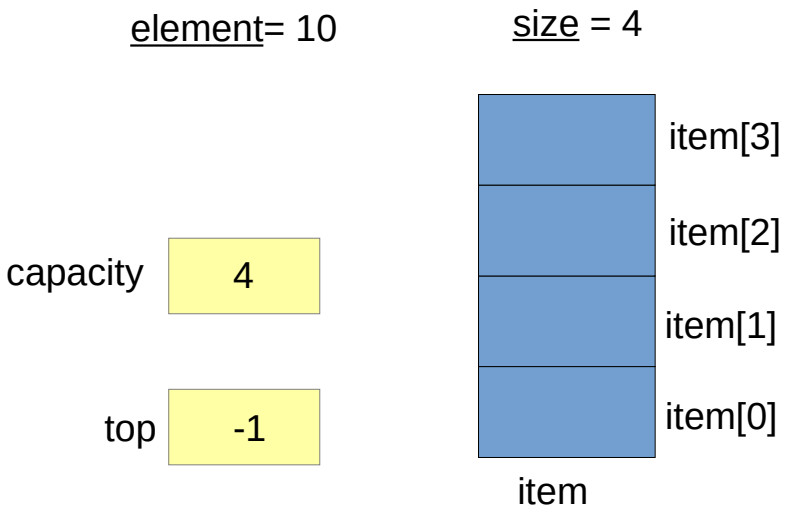
element : item to be added

**Output Specification:**

Status : e_true / e_false

Data Structure –Stack-Array Implementation

# push(stack,element)



Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
      return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```

element= 10

size = 4

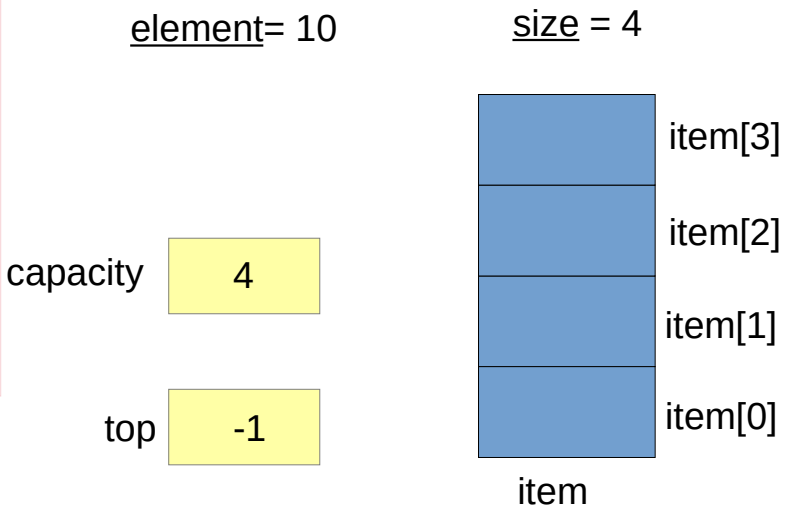
capacity 4

top -1

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
      return e_false
 ++stack⟶top
 stack⟶item[stack⟶top] = element
  return e_true;
```




is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

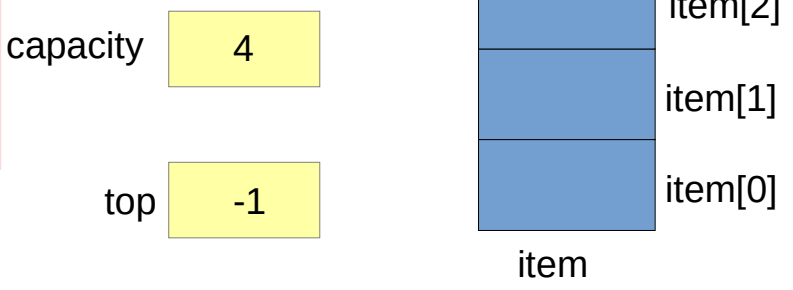
Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```

element= 10

size = 4



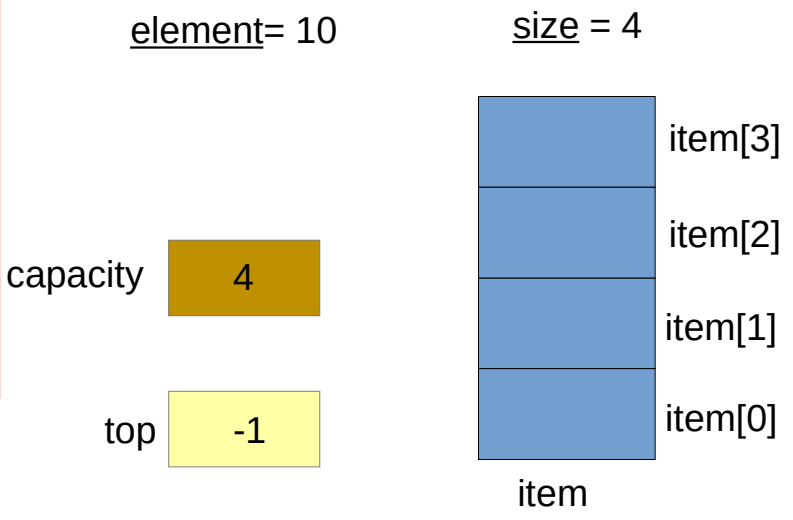

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```




is_stack_full(stack)

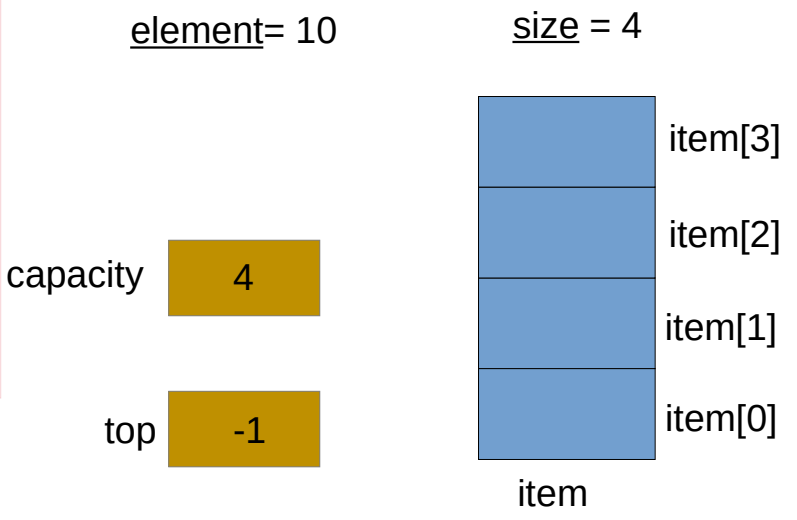
```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```




is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

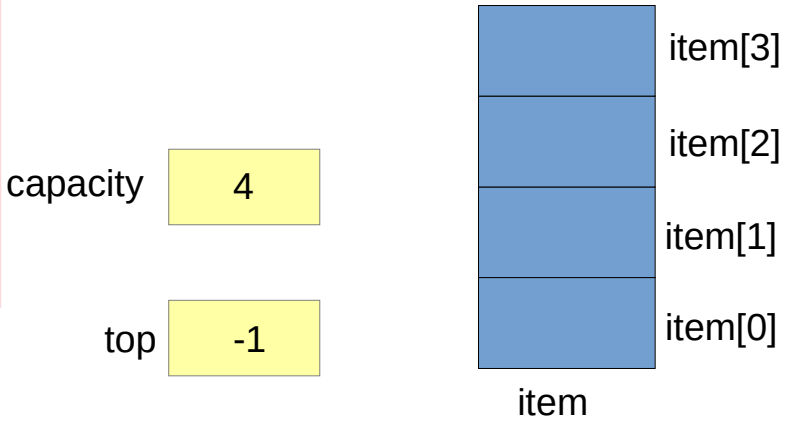
Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```

element= 10

size = 4



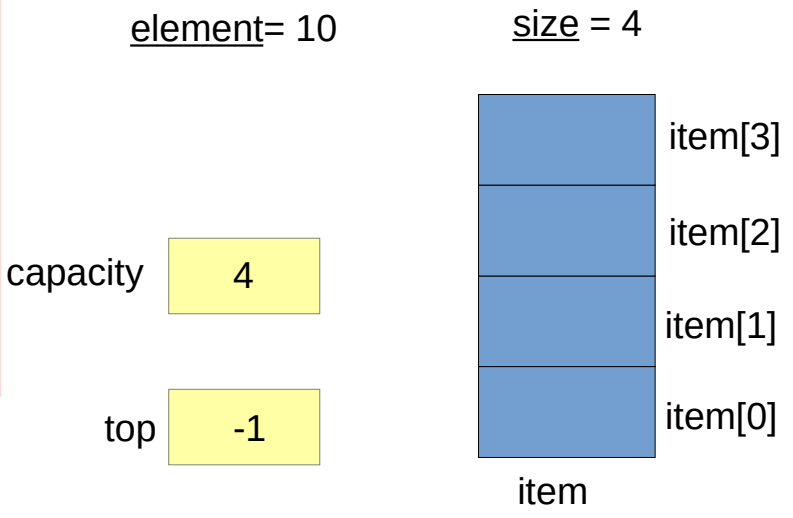

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
  return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
      return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```




is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```

element= 10

size = 4

capacity 4

top -1


is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```

element= 10

size = 4

capacity 4

top 0


is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
return e_true;
```

element= 10

size = 4

capacity 4

top 0


is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

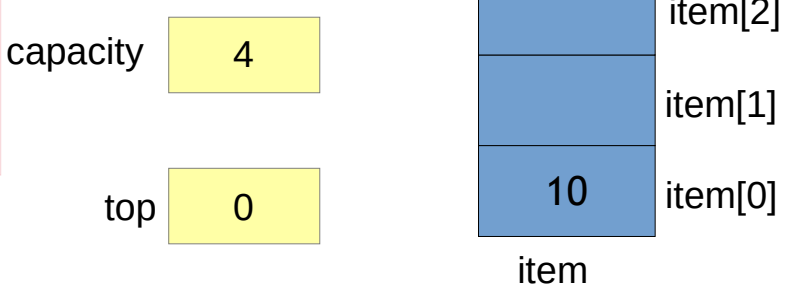
Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
return e_true;
```

element= 10

size = 4




is_stack_full(stack)

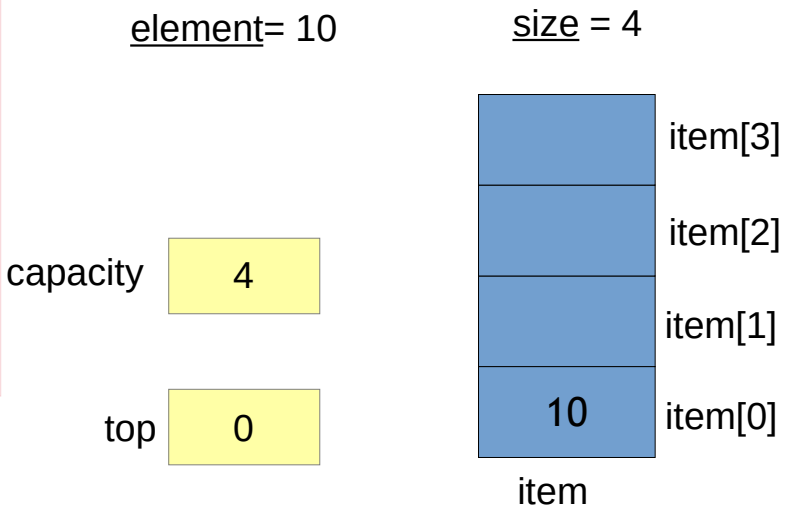
```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
return e_true;
```



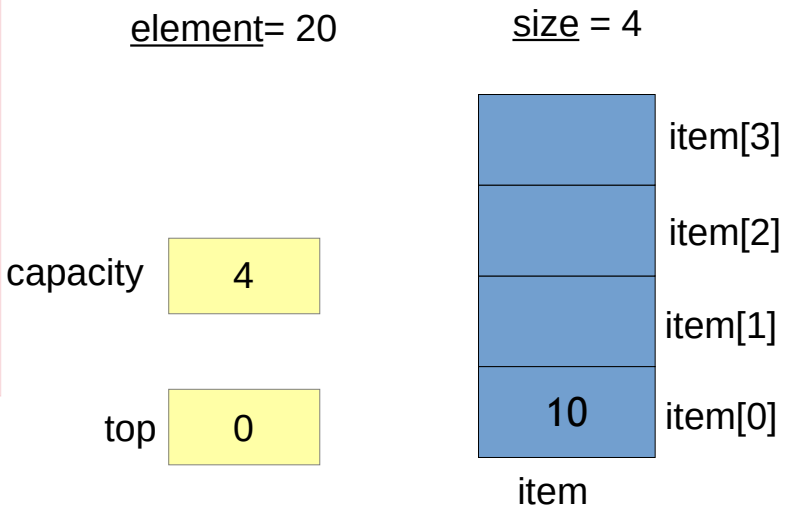

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```



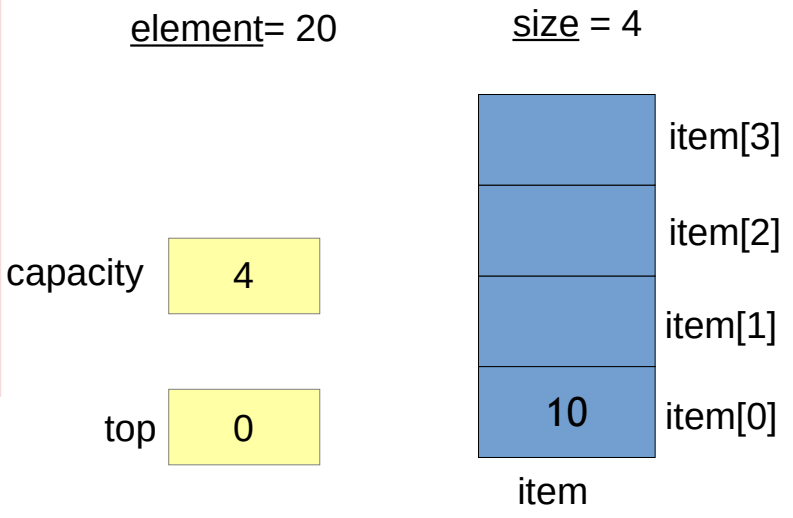

## is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
      return e_false
 ++stack⟶top
 stack⟶item[stack⟶top] = element
  return e_true;
```



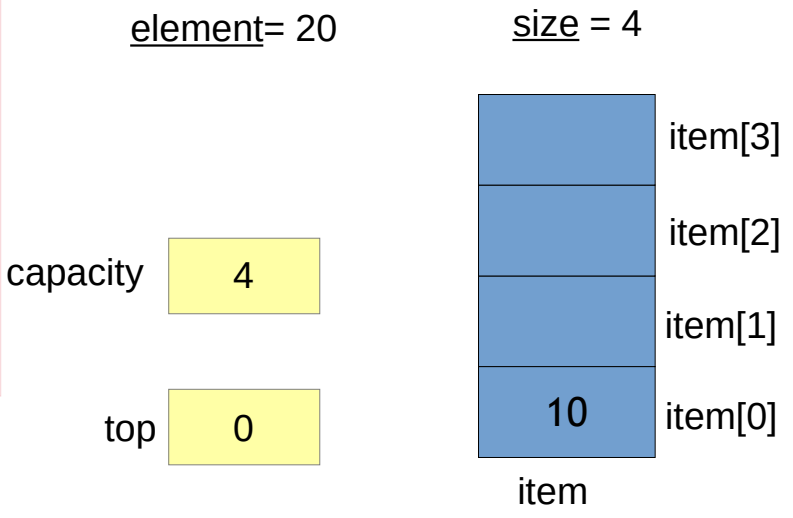

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```




is_stack_full(stack)

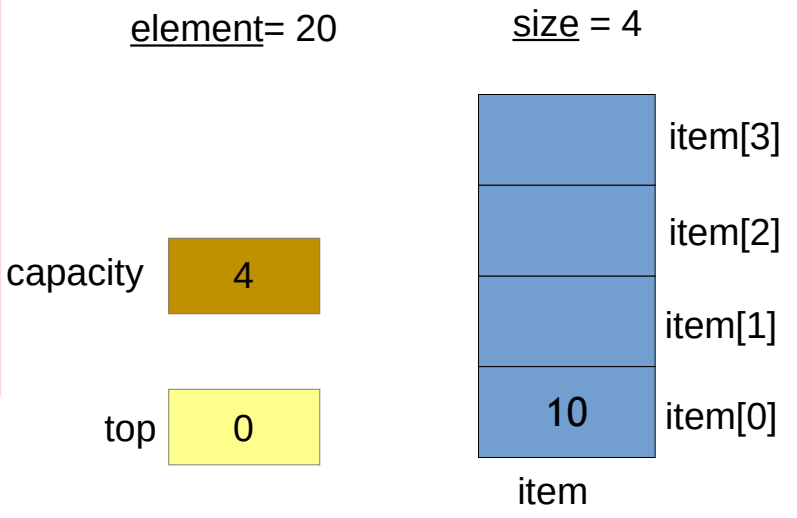
```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```




is_stack_full(stack)

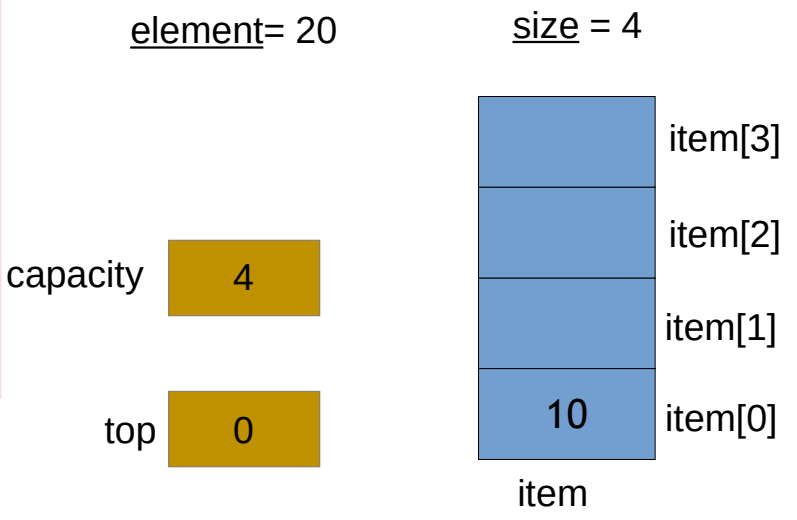
```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```



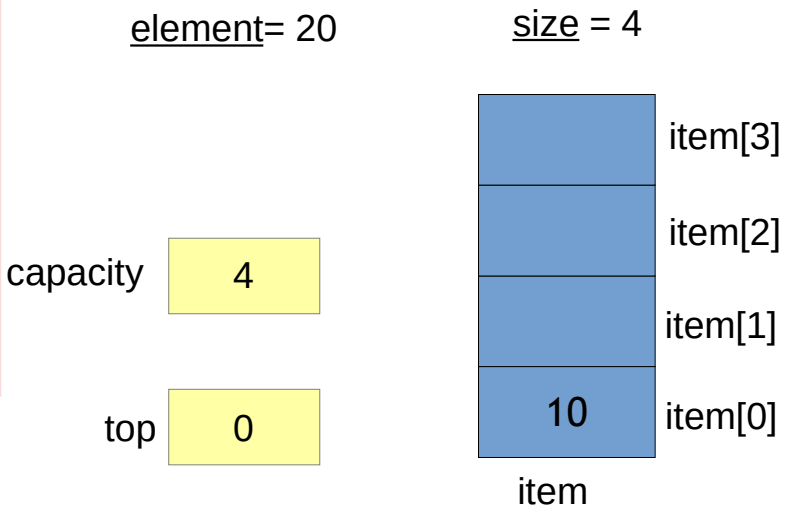

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```




is_stack_full(stack)

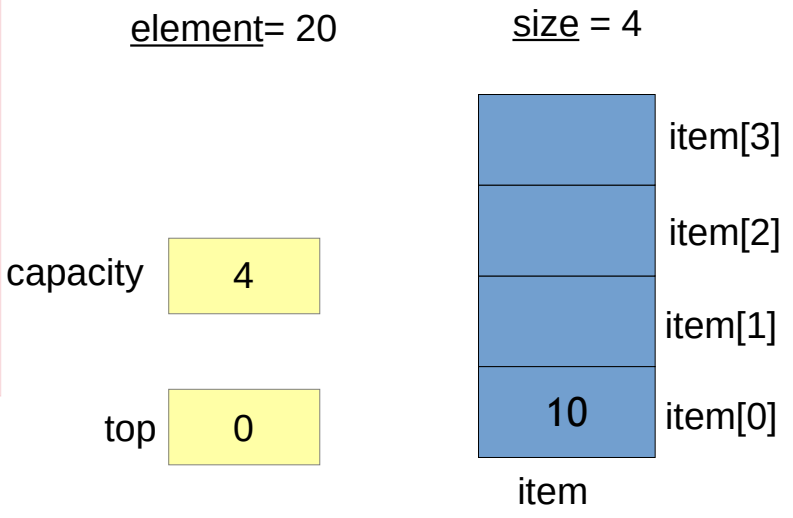
```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
  return e_false
```

EMERTXE

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
      return e_false
 ++stack⟶top
 stack⟶item[stack⟶top] = element
  return e_true;
```



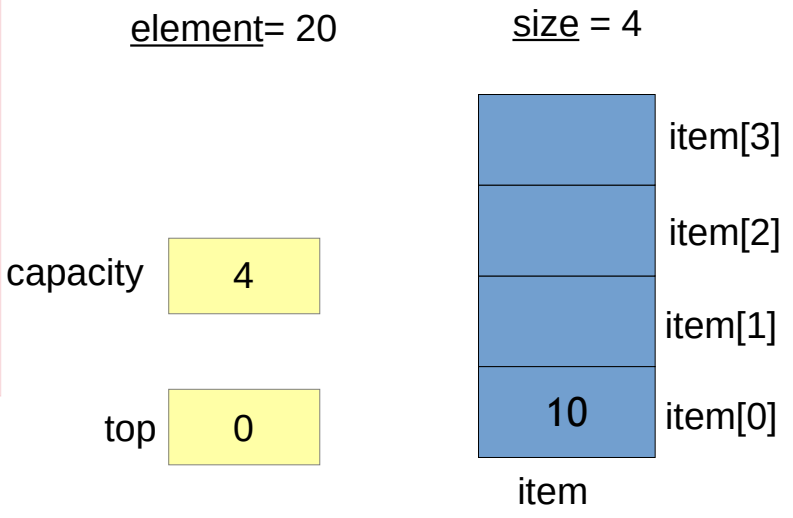

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```



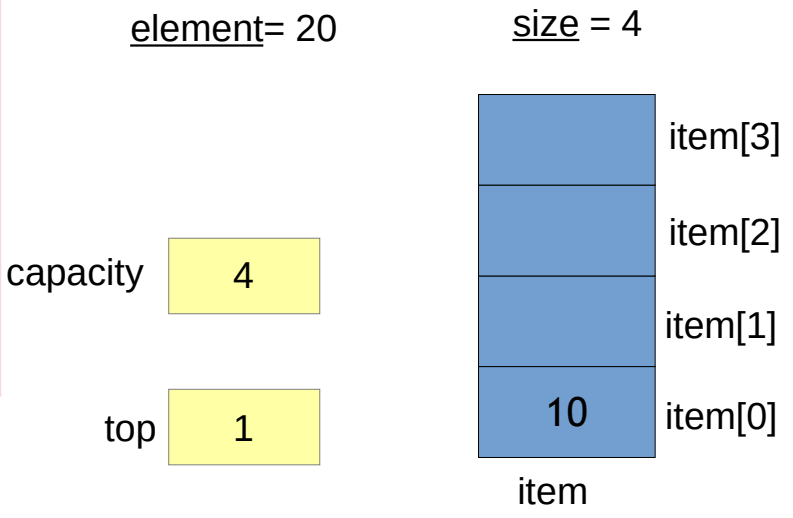

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```




is_stack_full(stack)

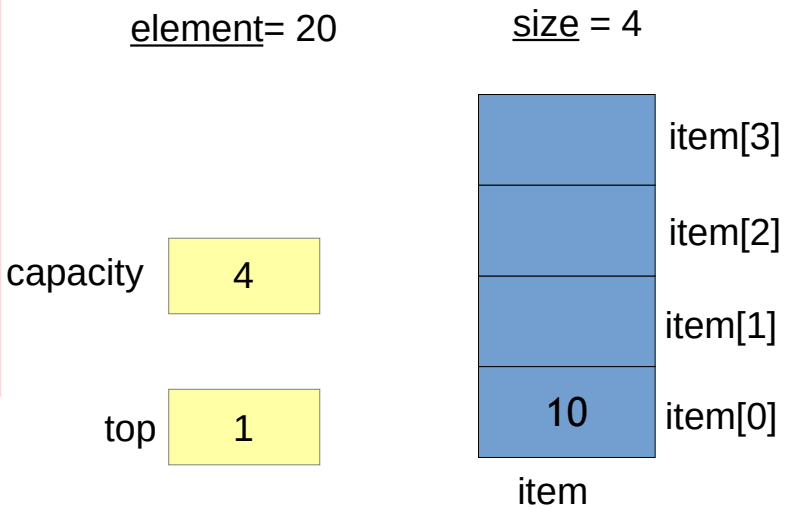
```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
return e_true;
```




is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

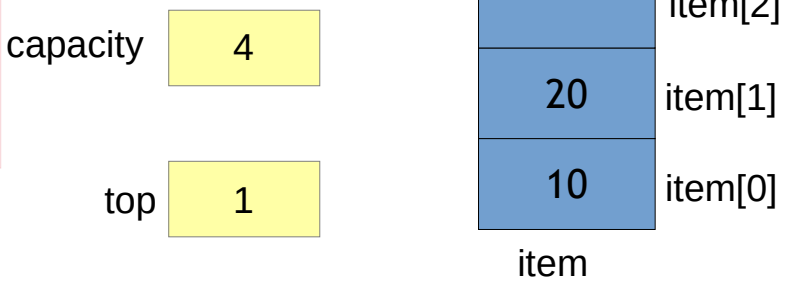
Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
return e_true;
```

element= 20

size = 4




is_stack_full(stack)

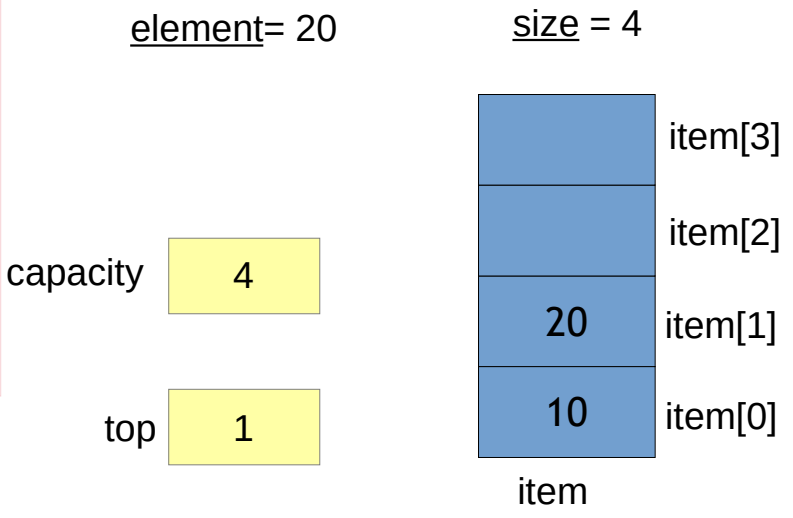
```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
return e_true;
```



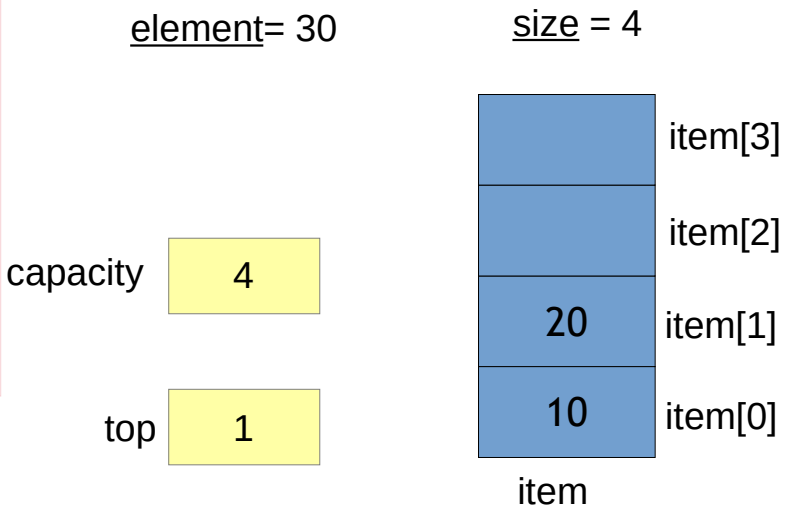

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```



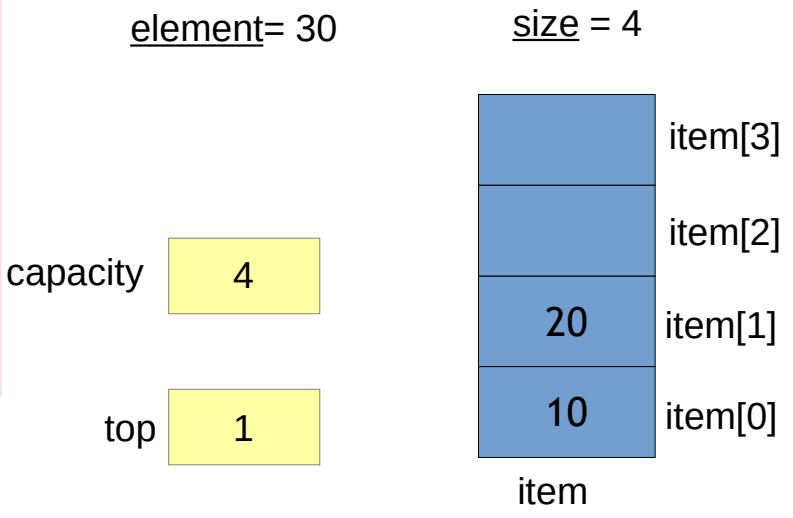

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
      return e_false
 ++stack⟶top
 stack⟶item[stack⟶top] = element
  return e_true;
```



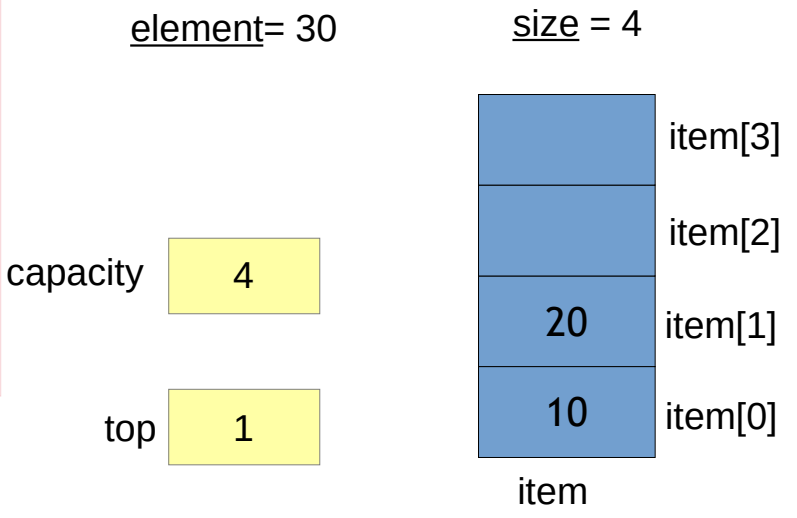

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```




is_stack_full(stack)

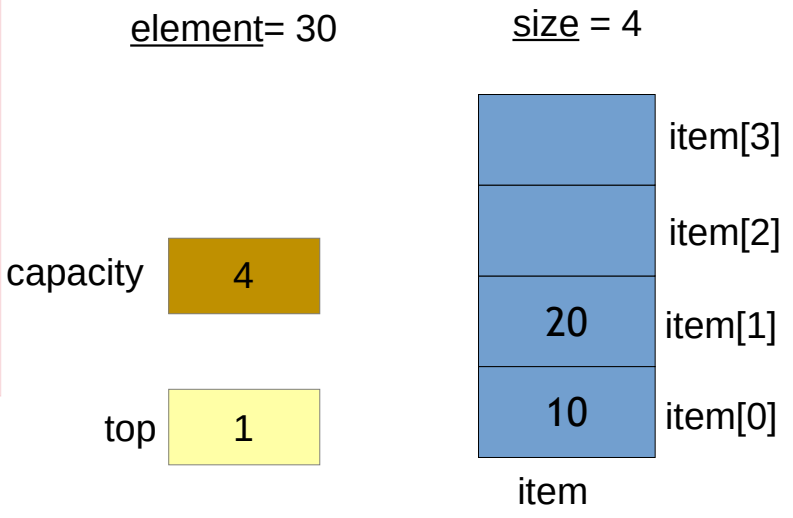
```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```




is_stack_full(stack)

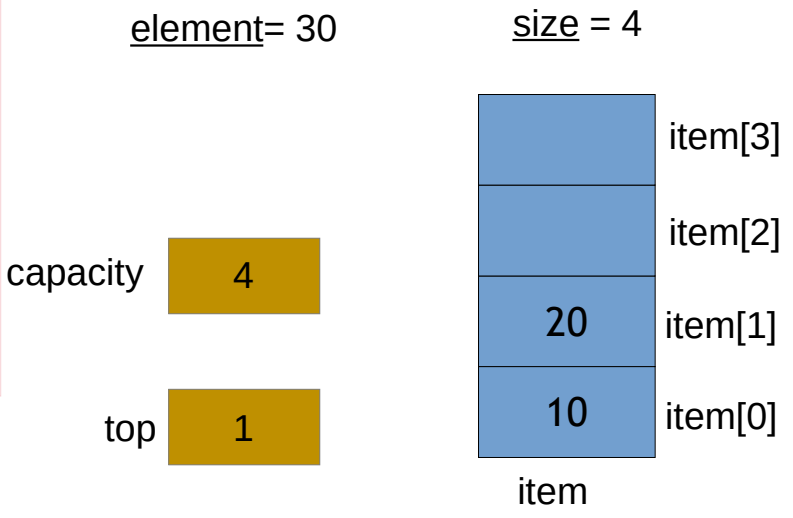
```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```



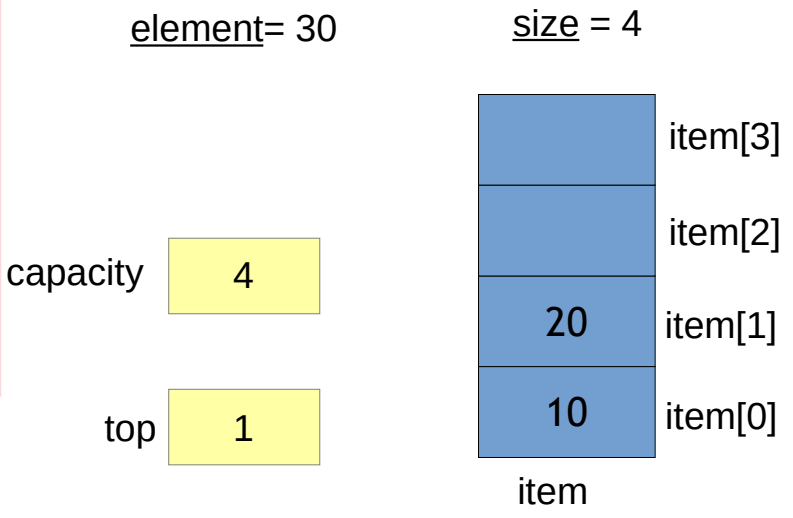

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```



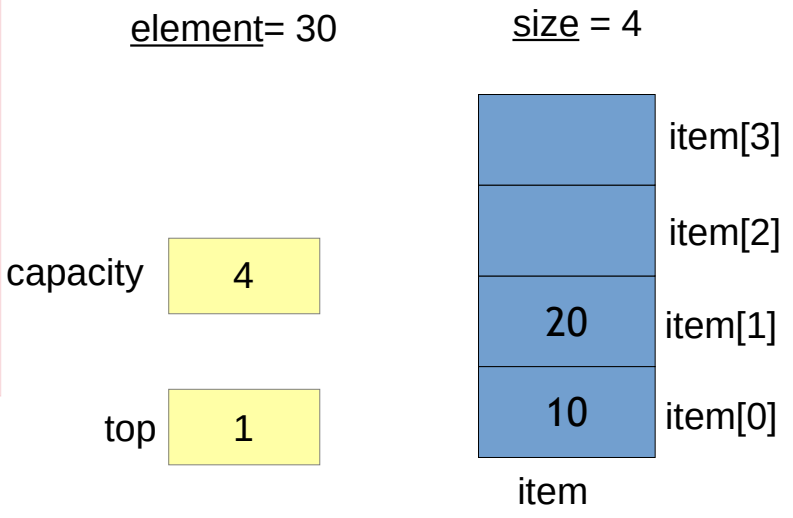

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
  return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
      return e_false
 ++stack⟶top
 stack⟶item[stack⟶top] = element
  return e_true;
```



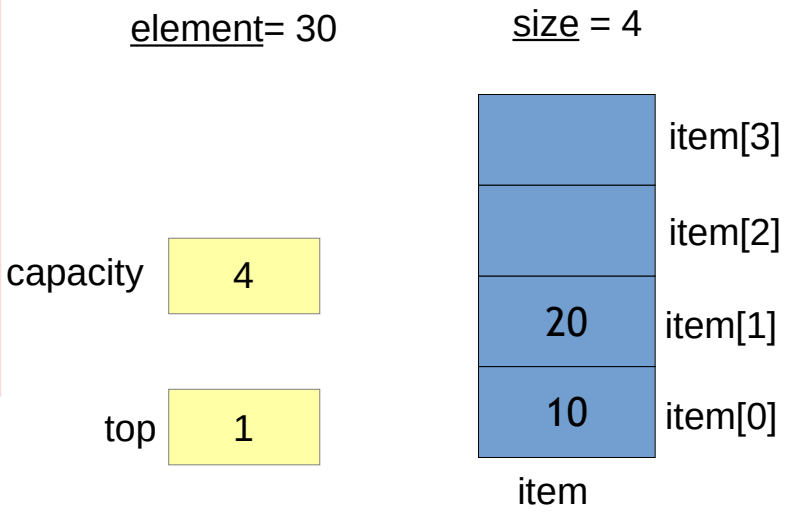

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```




## is_stack_full(stack)

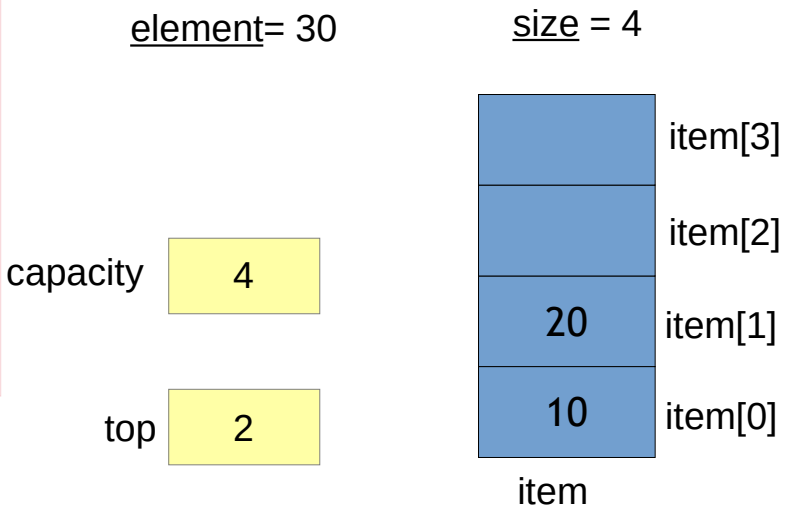
```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```




## is_stack_full(stack)

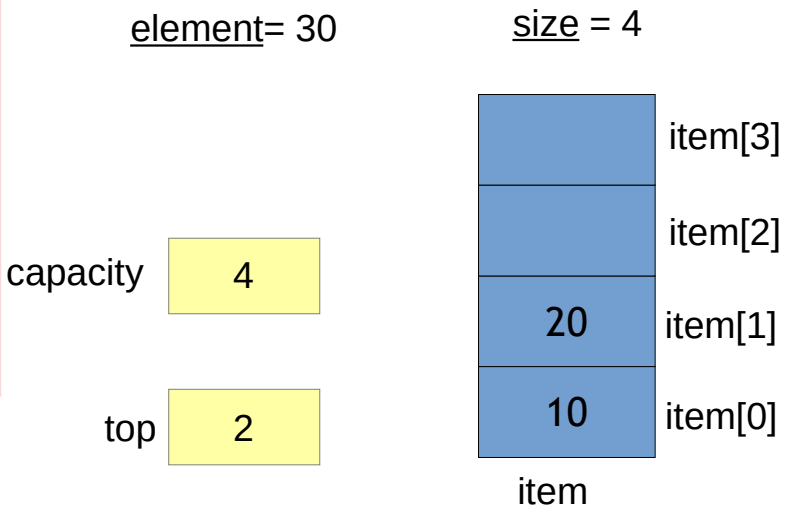
```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
return e_true;
```




is_stack_full(stack)

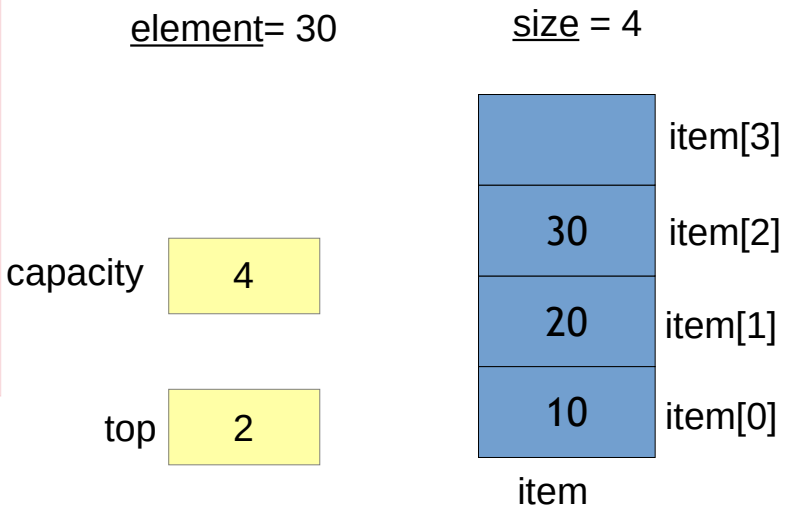
```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
return e_true;
```



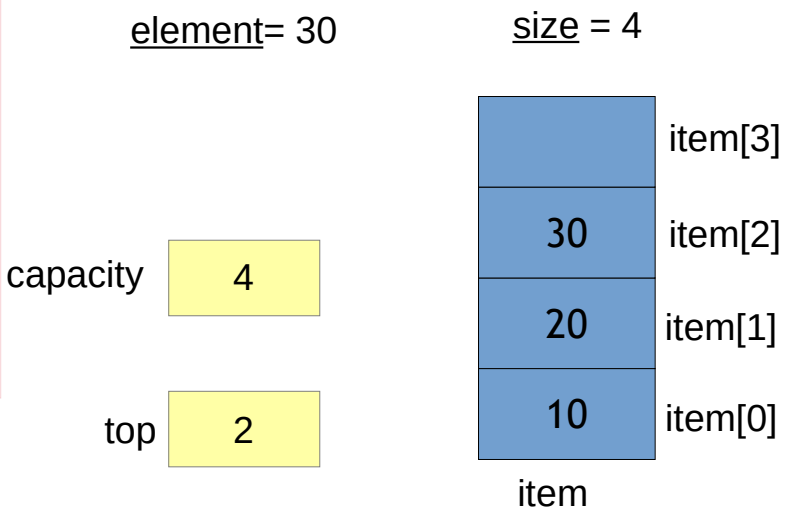

## is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
return e_true;
```



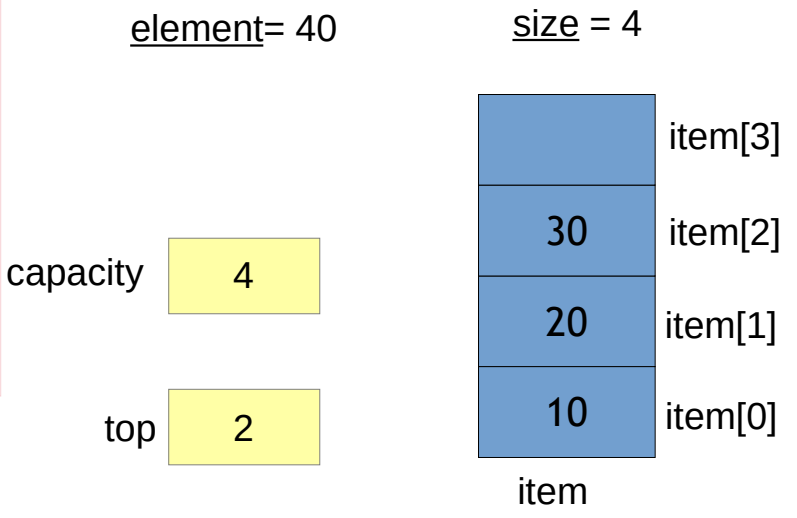

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```



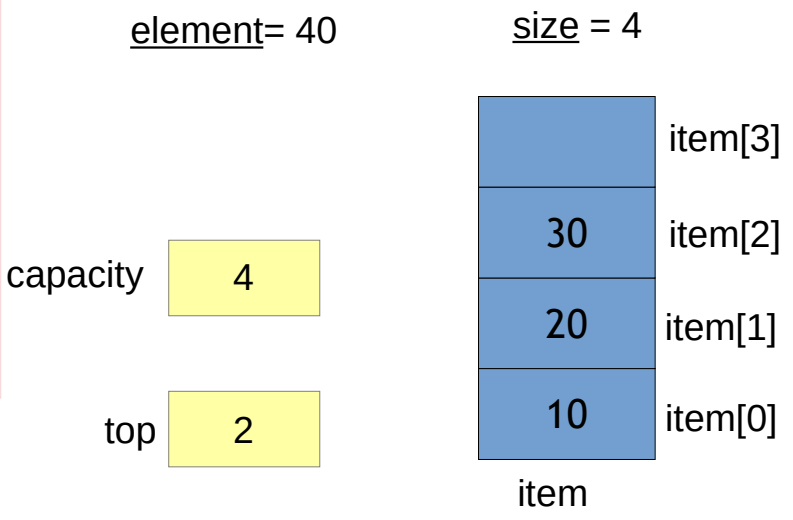

## is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
      return e_false
 ++stack⟶top
 stack⟶item[stack⟶top] = element
  return e_true;
```



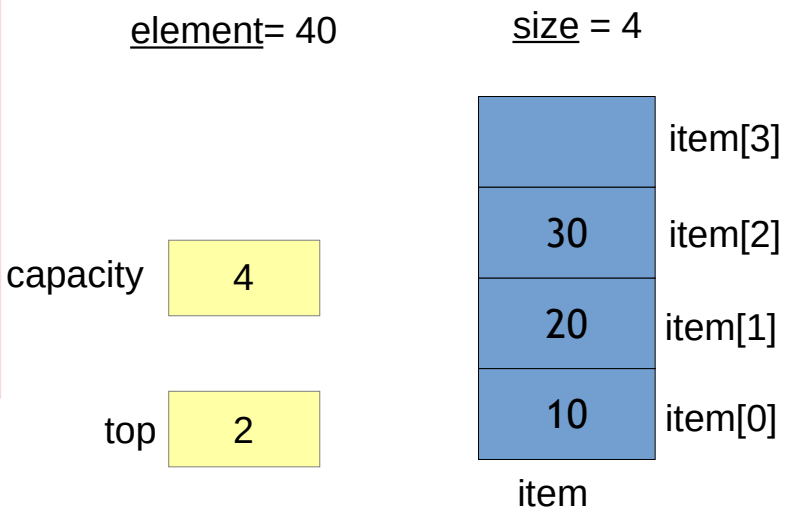

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```



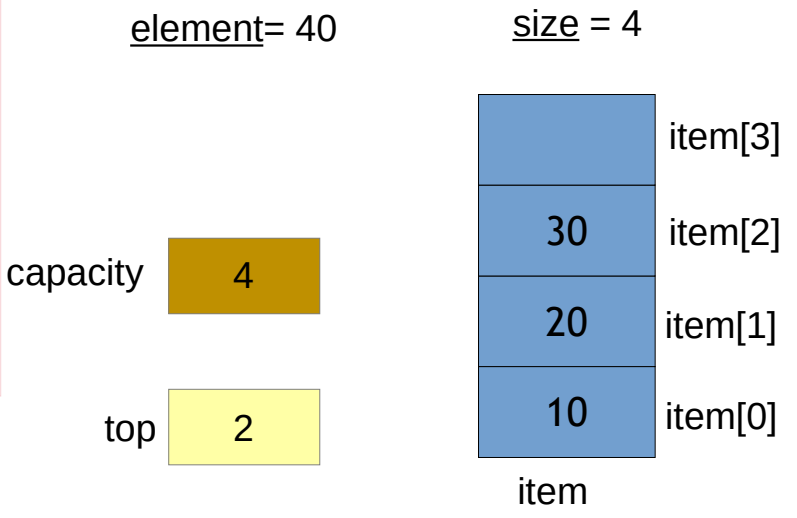

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```



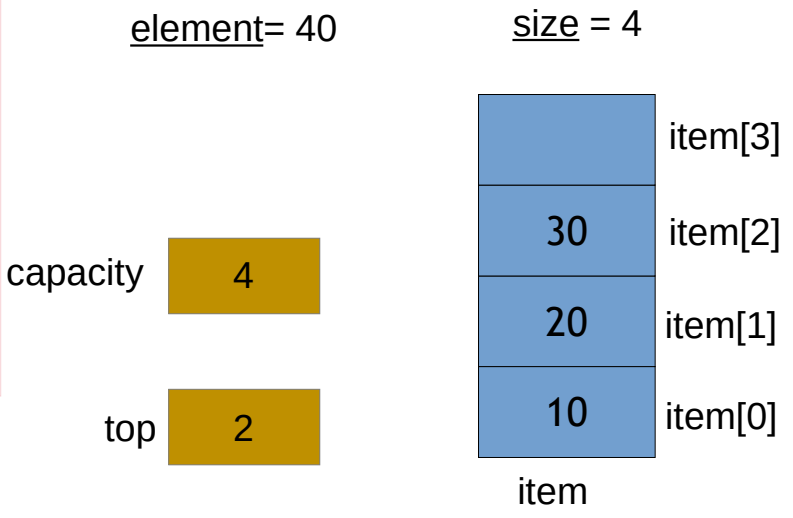

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```




is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

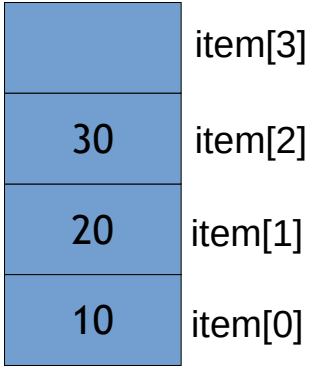
```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```

element= 40

size = 4

capacity 4

top 2




is_stack_full(stack)

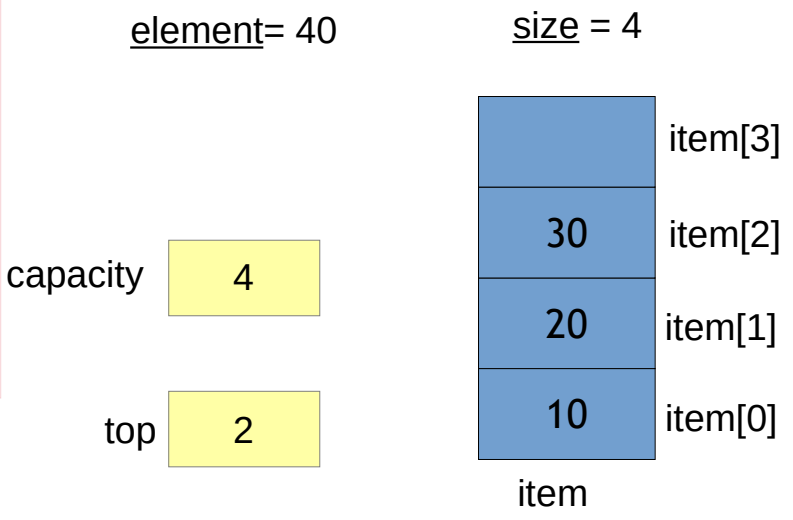
```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
      return e_false
 ++stack⟶top
 stack⟶item[stack⟶top] = element
  return e_true;
```



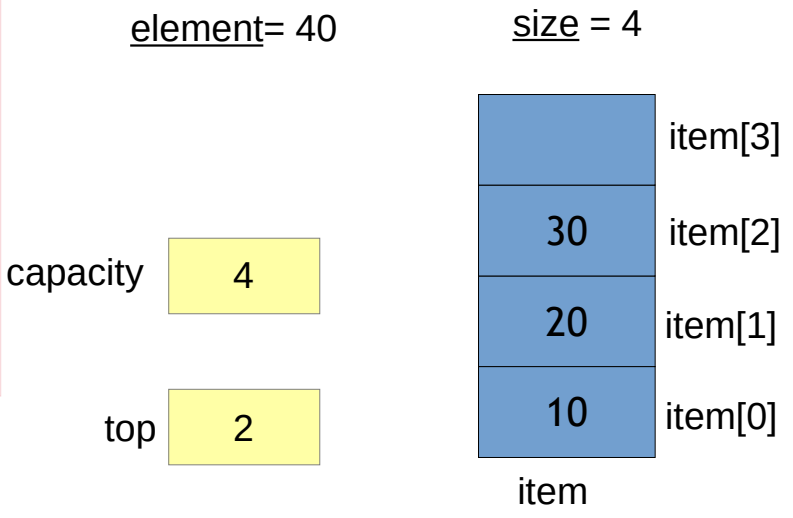

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```



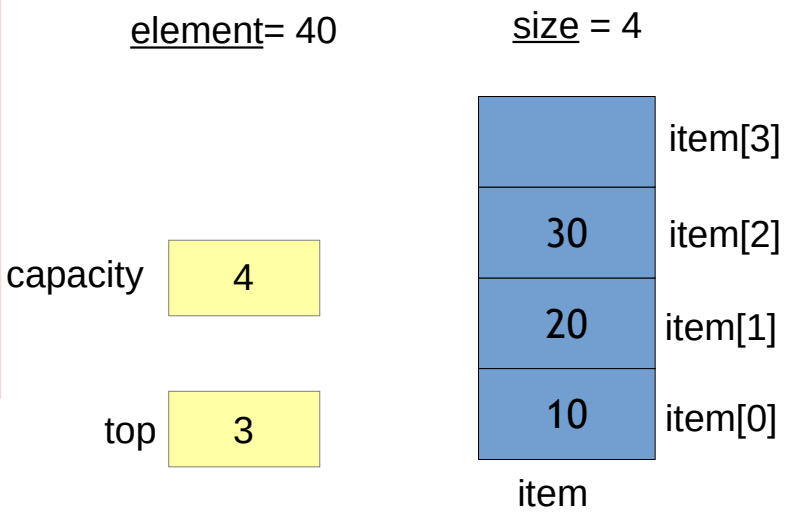

## is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```



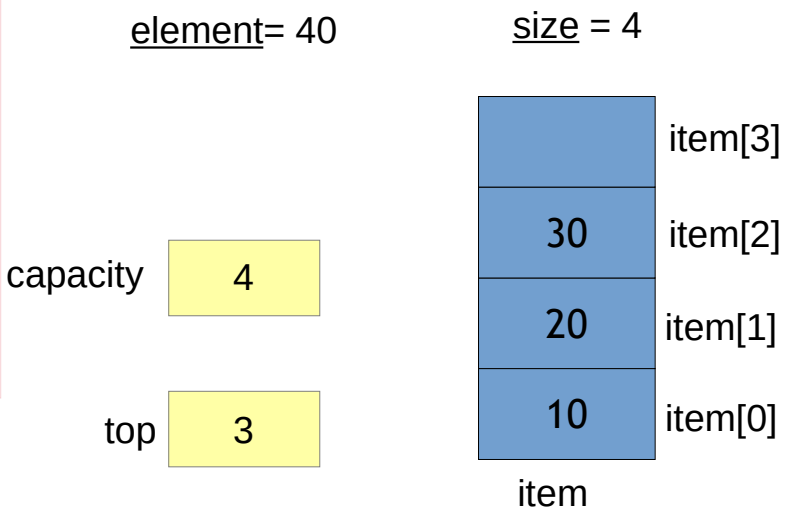

## is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
return e_true;
```



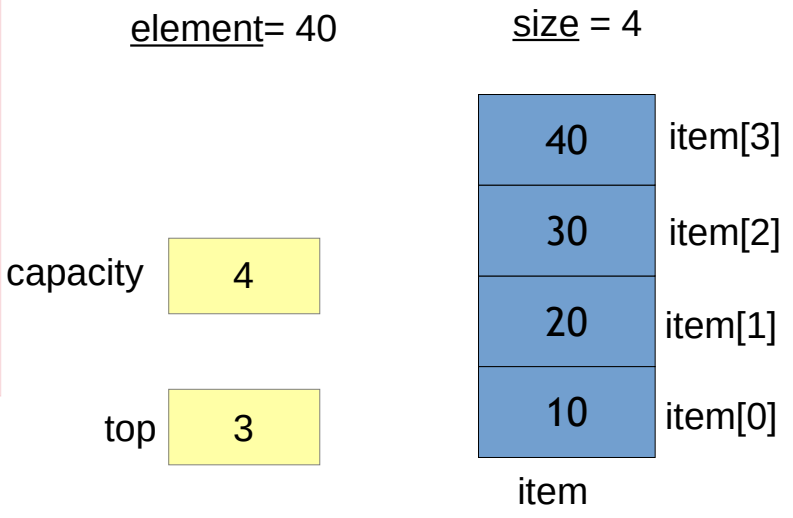

## is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
return e_true;
```



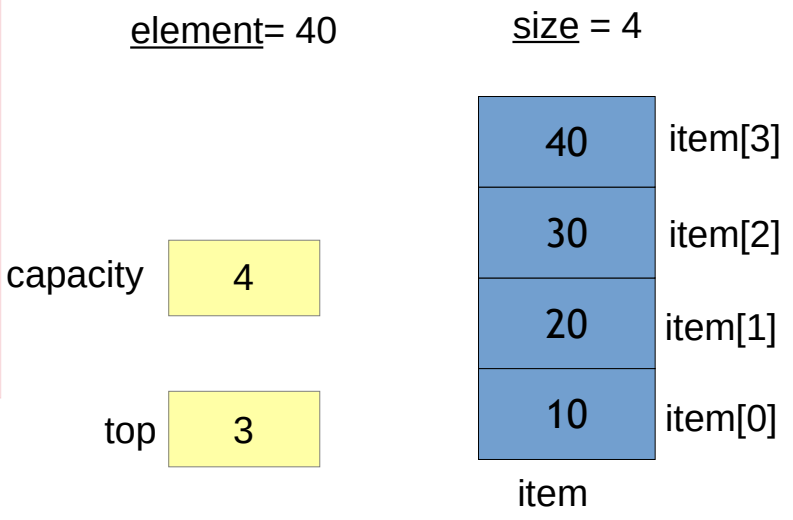

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
return e_true;
```



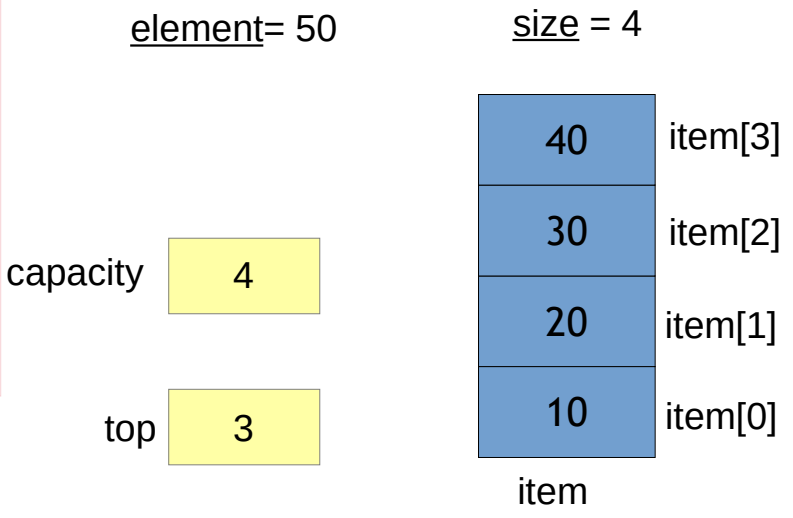

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```




## is_stack_full(stack)

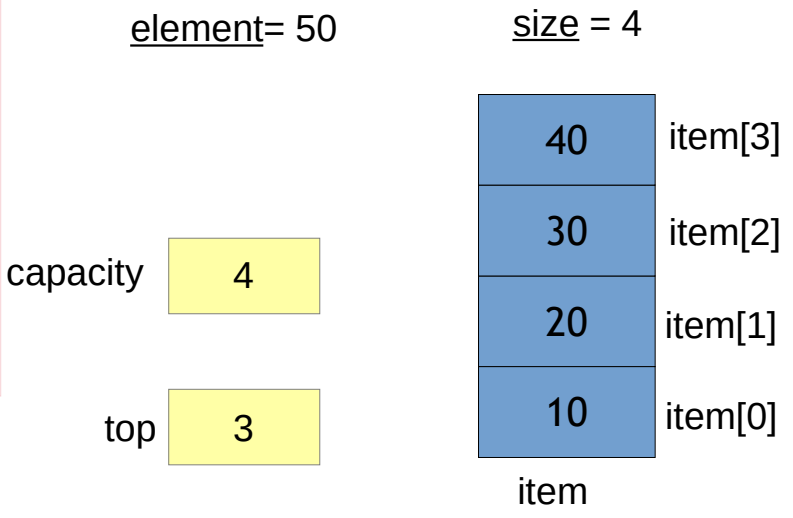
```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

EMERTXE

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
      return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```



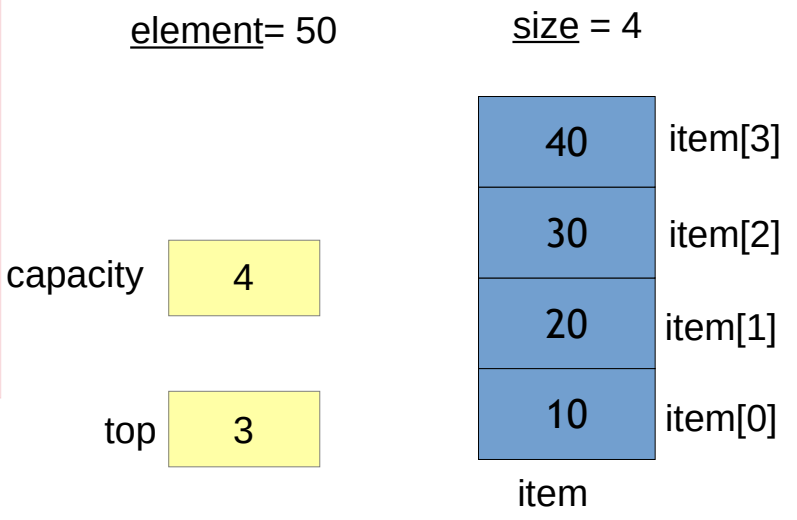

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```



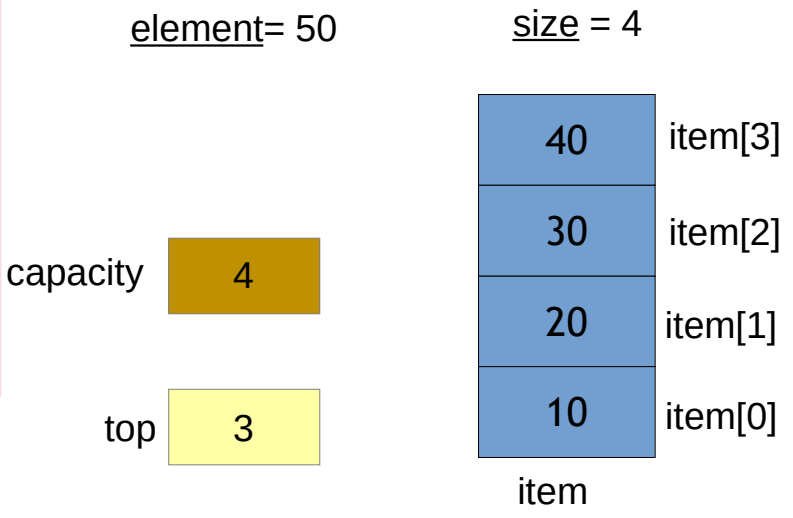

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
     return e_true
else
    return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```



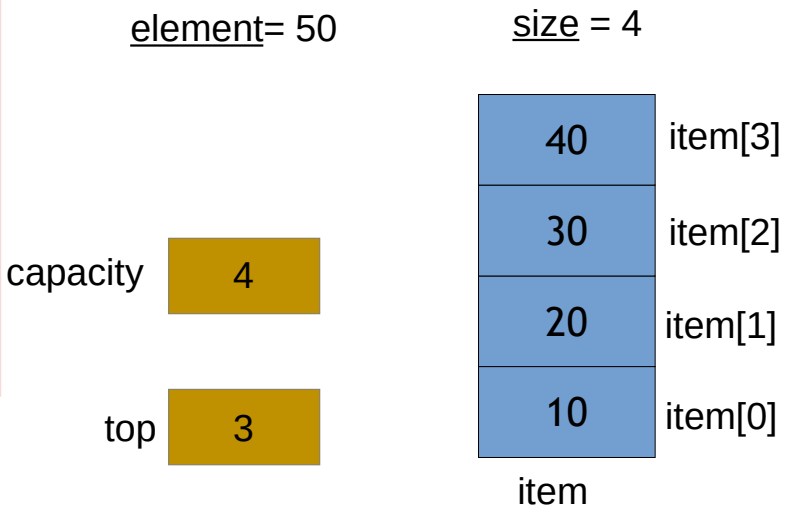

is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
     return e_true
else
    return e_false
```

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```

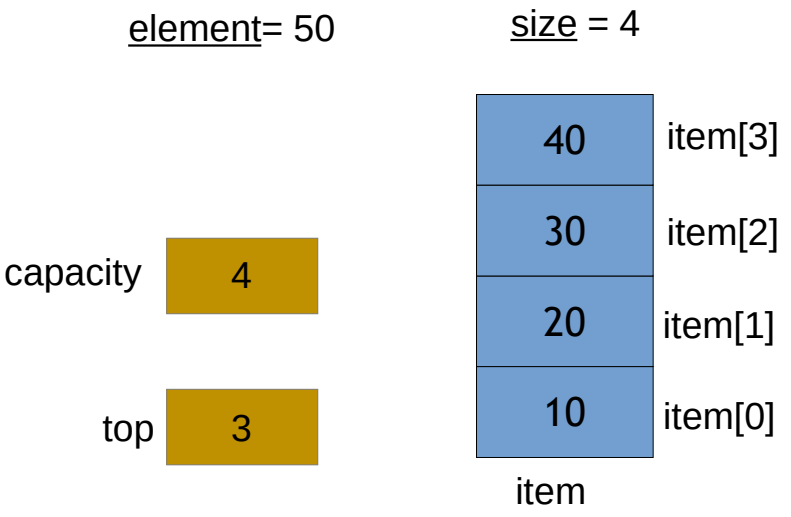



is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
     return e_true
else
    return e_false
```

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```




is_stack_full(stack)

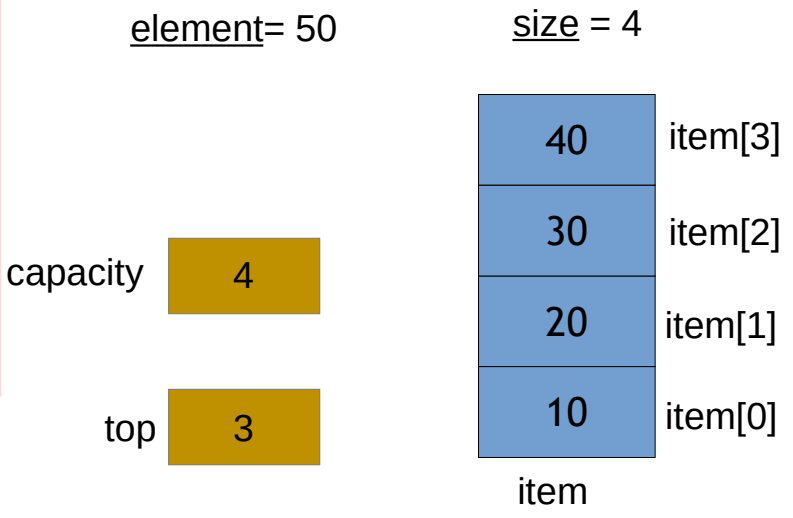
```
If (stack⟶top = stack⟶capacity -1)
     return e_true
else
    return e_false
```

Stack is full

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
     return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```




is_stack_full(stack)

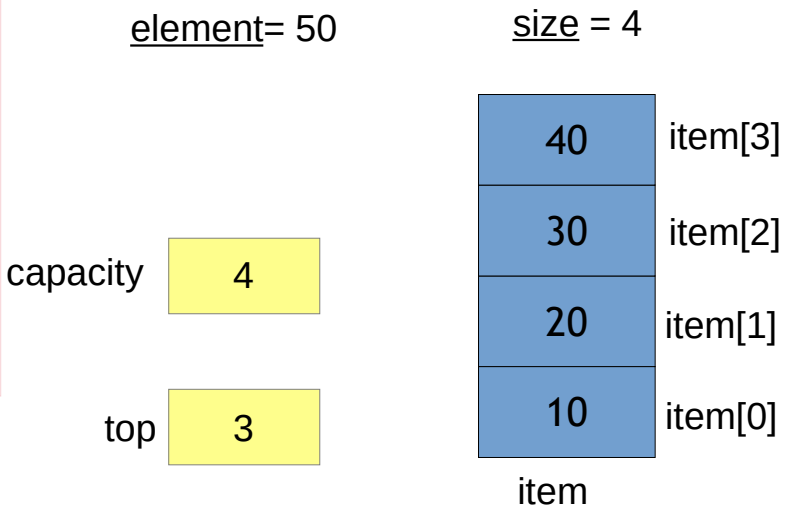
```
If (stack⟶top = stack⟶capacity -1)
   return e_true
else
   return e_false
```

Stack is full

Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
      return e_false
 ++stack⟶top
 stack⟶item[stack⟶top] = element
  return e_true;
```




is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Stack is full

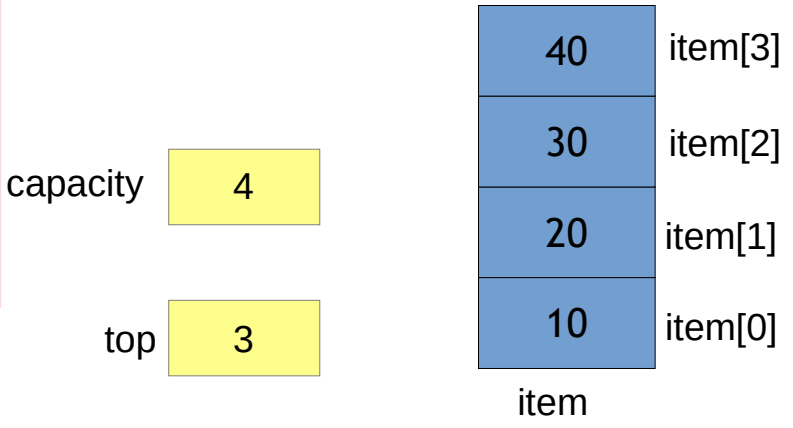
Data Structure –Stack-Array Implementation

# push(stack,element)

```
If (is_stack_full(stack))
    return e_false
++stack⟶top
stack⟶item[stack⟶top] = element
 return e_true;
```

element= 50

size = 4




## is_stack_full(stack)

```
If (stack⟶top = stack⟶capacity -1)
    return e_true
else
   return e_false
```

Stack is full

EMERTXE

# Stack – pop(stack,element)