Εισαγωγή στην πληροφορική και τις εφαρμογές της

ΠΑΝΑΓΙΩΤΗΣ ΠΑΠΑΖΟΓΛΟΥ

Website: https://papazoglou-files.gr/books/

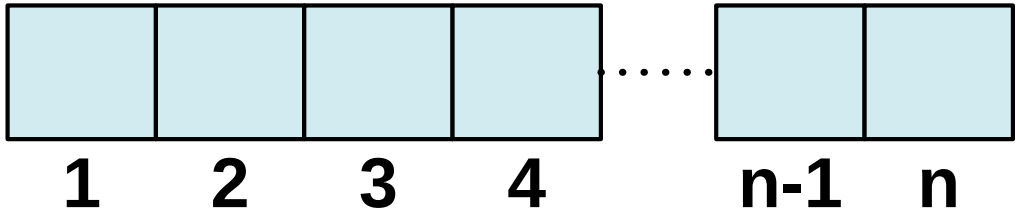
Επιστημονικές Εκδόσεις ΤΖΙΟΛΑ

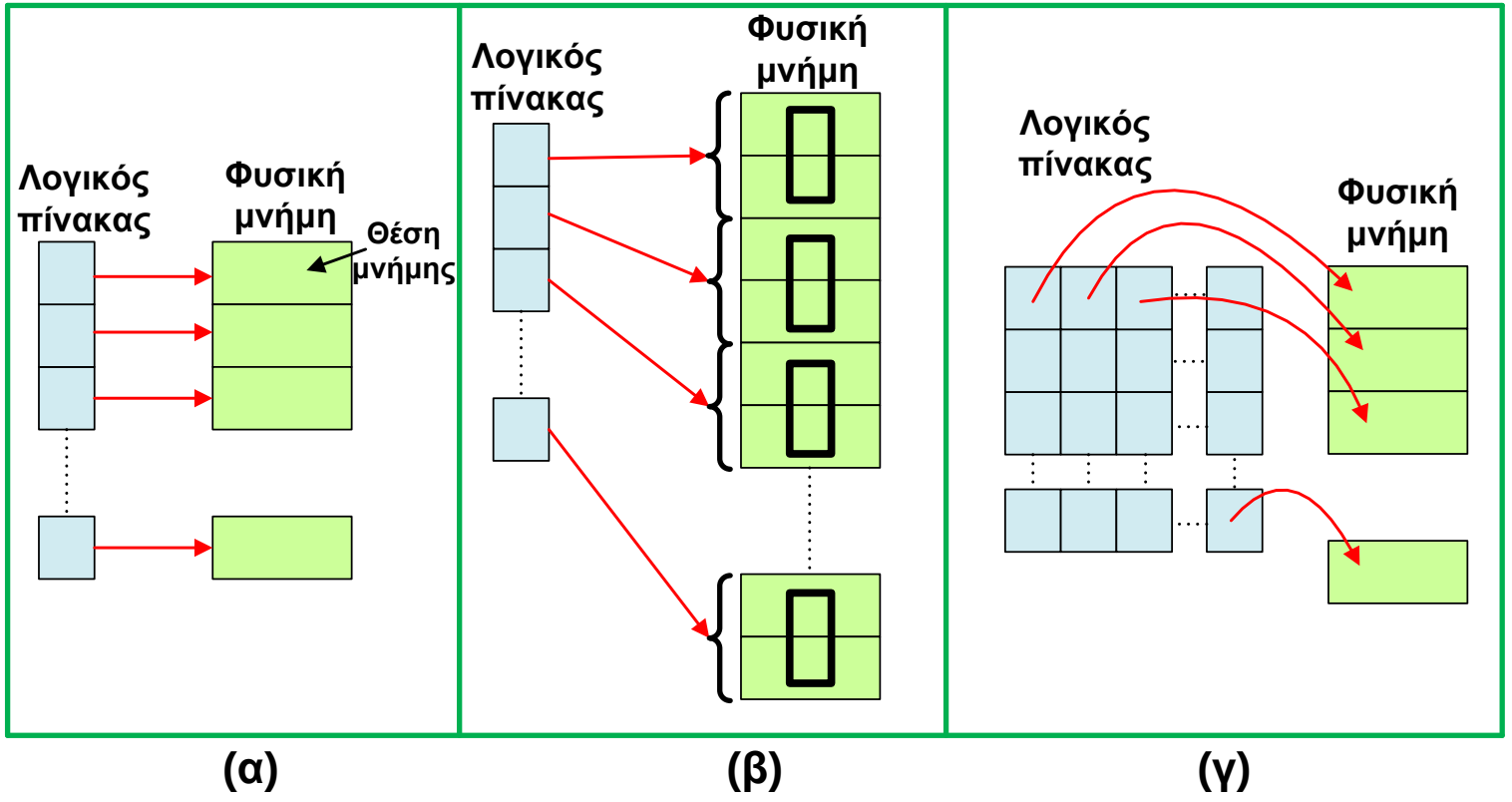
ΚΕΦΑΛΑΙΟ 16

Βασικές δομές δεδομένων και αλγόριθμοι

Επιστημονικές Εκδόσεις ΤΖΙΟΛΑ

Πίνακες
Μονοδιάστατοι Πίνακες (1)





Πίνακες
Μονοδιάστατοι Πίνακες (2)

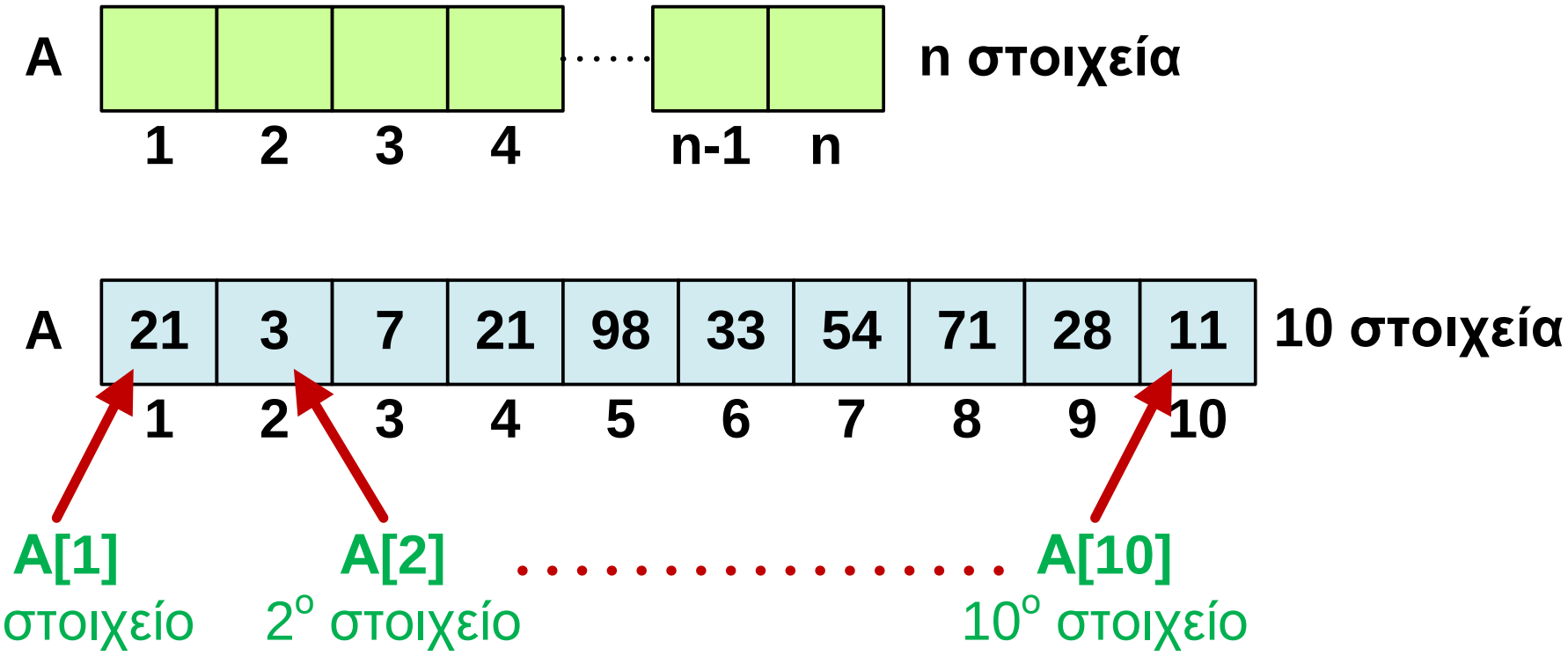
Πλεονεκτήματα

- Εύκολη προσπέλαση και δημιουργία
- Τυχαία προσπέλαση (απευθείας πρόσβαση σε οποιοδήποτε στοιχείο)
- Αποδοτική χρήση αλγορίθμων διαχείρισης και επεξεργασίας (π.χ. αναζήτηση, εύρεση μικρότερου στοιχείου, κλπ)

Μειονεκτήματα

- Το μέγεθος του πίνακα είναι σταθερό
- Φιλοξενεί στοιχεία του ίδιου τύπου
- Η παρεμβολή νέων στοιχείων είναι χρονοβόρα, αφού για να γίνει η αντίστοιχη εισαγωγή, θα πρέπει να ολισθήσουν μια θέση όλα τα υπόλοιπα στοιχεία

Πίνακες
Μορφή μονοδιάστατων πινάκων



Πίνακες

Αρχικοποίηση

Προσέγγιση με βρόχο ΓΙΑ	Προσέγγιση με βρόχο ΟΣΟ
Για x από 1 έως n, βήμα 1 A[x]=0 Τέλος-Για	x=1 ΟΣΟ x<=n ΕΠΑΝΕΛΑΒΕ A[x]=0 x=x+1 Τέλος-ΟΣΟ

Γέμισμα πίνακα

Για x από 1 έως n, βήμα 1
Διάβασμα A[x]
Τέλος-Για

Πίνακες

Εμφάνιση περιεχομένου του πίνακα

```
Για x από 1 έως n, βήμα 1
        Εμφάνιση A[x]
Τέλος-Για
```

Προηγούμενο και επόμενο στοιχείο

Αν η τρέχουσα θέση του δείκτη είναι **x=k**, τότε, εφόσον υπάρχει επόμενο στοιχείο, αυτό είναι το **A[k+1]**, ενώ το προηγούμενο, εφόσον υπάρχει, είναι το **A[k-1]**

Ελάχιστο στοιχείο

```
MIN=A[1]
Για x από 2 έως n, βήμα 1
            Αν A[x]<MIN ΤΟΤΕ
    MIN=A[x]
Τέλος-Για
```

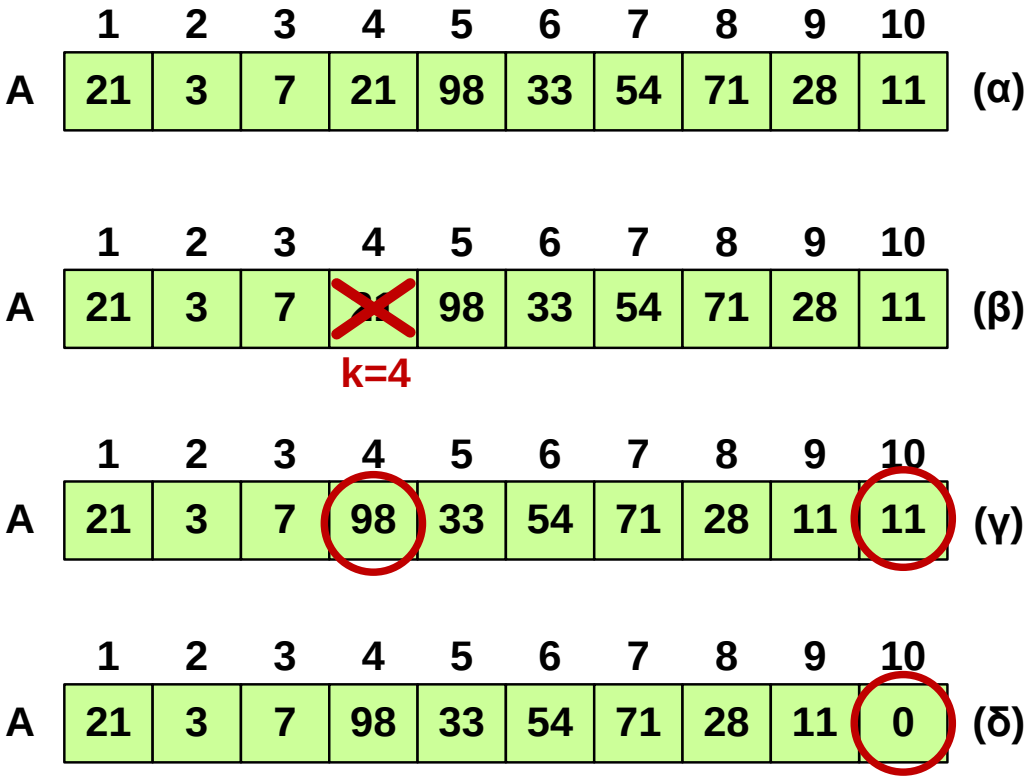
Πίνακες
Εύρεση στοιχείου

```
Διάβασμα στοιχείο_προς_έυρεση
Για x από 1 έως n, βήμα 1
            Αν A[x]= στοιχείο_προς_έυρεση ΤΟΤΕ
      Εμφάνιση «Βρέθηκε στη θέση»,x
      Έξοδος από το βρόχο
Τέλος-Για
```

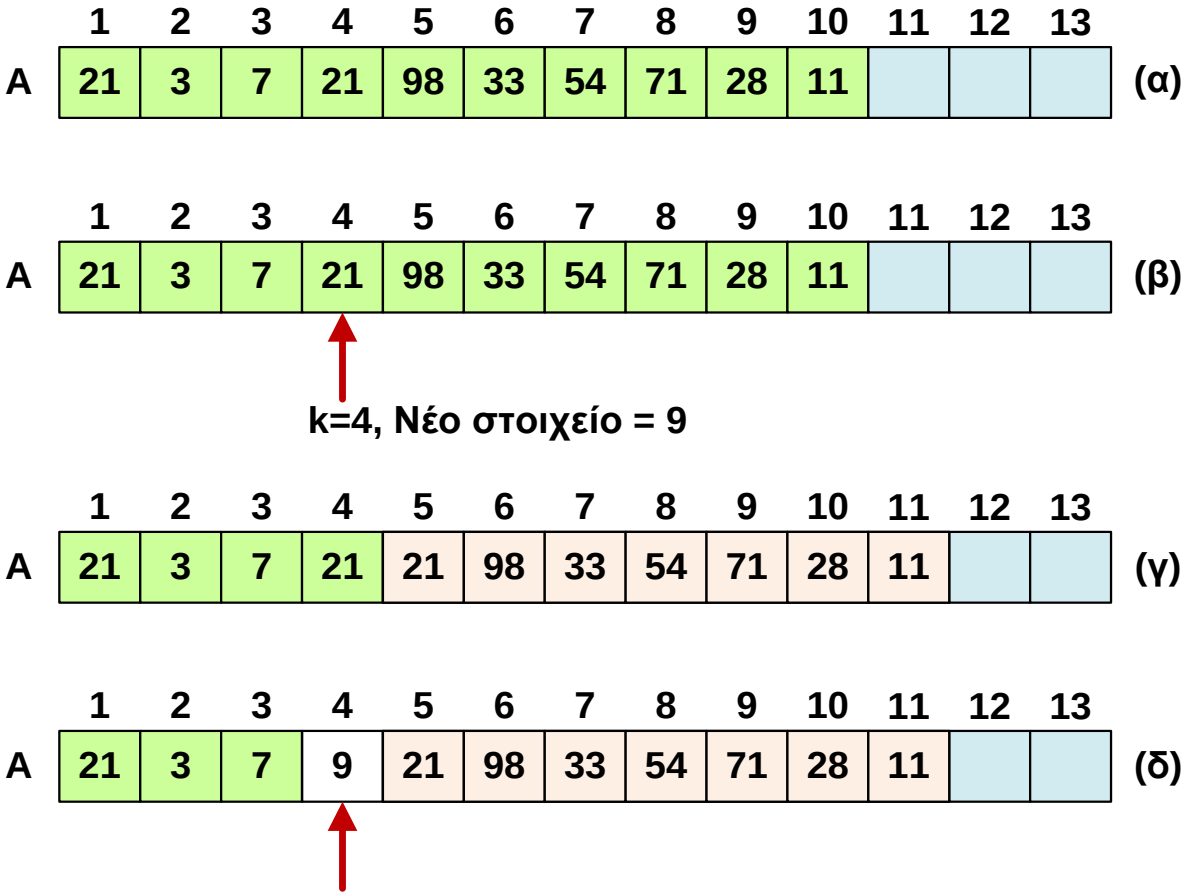
```
Διάβασμα στοιχείο_προς_εύρεση
FOUND=0
x=1
ΟΣΟ FOUND=0 ΕΠΑΝΕΛΑΒΕ
            Αν A[x]=στοιχείο_προς_εύρεση
                        Εμφάνιση «Βρέθηκε στη θέση», x
                        FOUND=1
            x=x+1
Τέλος-ΟΣΟ
```

Πίνακες

Διαγραφή στοιχείου

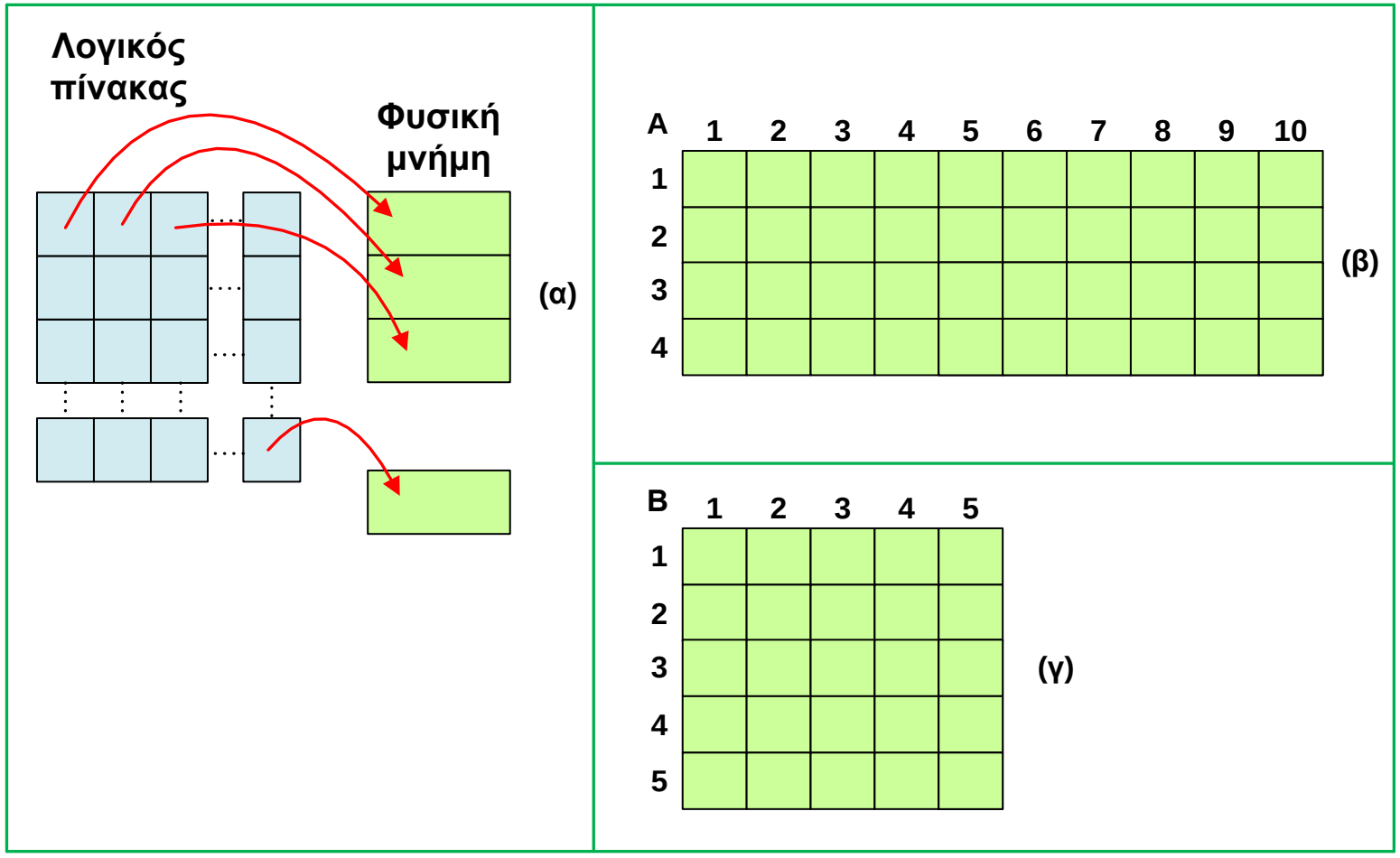


Εισαγωγή στοιχείου



Πίνακες

Πίνακες δύο διαστάσεων



Πίνακες

Πίνακες δύο διαστάσεων - Γέμισμα

```
Για line από 1 έως i
    Για column από 1 έως j
            Διάβασε στοιχείο A[line][column]
    Τέλος-Για
Τέλος-Για
```

Πίνακες - Βασικοί υπολογισμοί, Αναζήτηση και ταξινόμηση

Άθροισμα και μέσος όρος

ΑΡΧΗ
Δεδομένα: Πίνακας Α, Ν θέσεων

/*Σχόλιο: Υπολογισμός μέσου όρου */
SUM=0, ΜΟ=0

Για i=1 έως Ν, βήμα 1
  Εμφάνιση Α[i] /*Σχόλιο: για βοηθητικούς λόγους */
  SUM=SUM+A[i]
Τέλος-Για

ΜΟ=SUM/N
Εμφάνιση ΜΟ
ΤΕΛΟΣ

Πίνακες - Βασικοί υπολογισμοί, Αναζήτηση και ταξινόμηση

Εμφάνιση και άθροισμα διαγωνίου

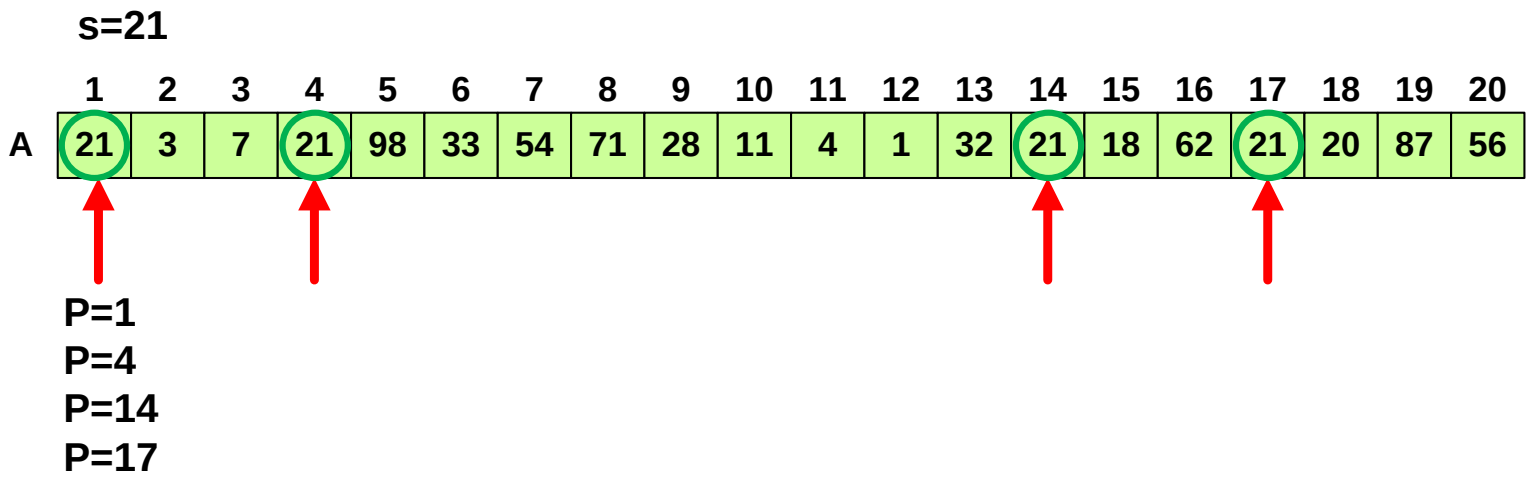
```
ΑΡΧΗ
Δεδομένα: Πίνακας Α, Ν x Ν
/*Σχόλιο: γέμισμα πίνακα */
Για i=1 έως Ν, βήμα 1
   Για j=1 έως Ν, βήμα 1
      Ανάγνωση x ή κλήρωση ενός τυχαίου αριθμού x
      A[i][j]=x
      Εμφάνιση A[i][j] /*Σχόλιο: για βοηθητικούς λόγους */
   Τέλος-Για
   Αλλαγή γραμμής
Τέλος-Για

/*Σχόλιο: εμφάνιση και άθροισμα διαγωνίου */
SUM=0
Για i=1 έως Ν, βήμα 1
   Εμφάνιση A[i][i]
   SUM = SUM + A[i][i]
Τέλος-Για
Εμφάνιση SUM
ΤΕΛΟΣ
```

Επιστημονικές Εκδόσεις ΤΖΙΟΛΑ

Πίνακες - Βασικοί υπολογισμοί, Αναζήτηση και ταξινόμηση

Σειριακή αναζήτηση

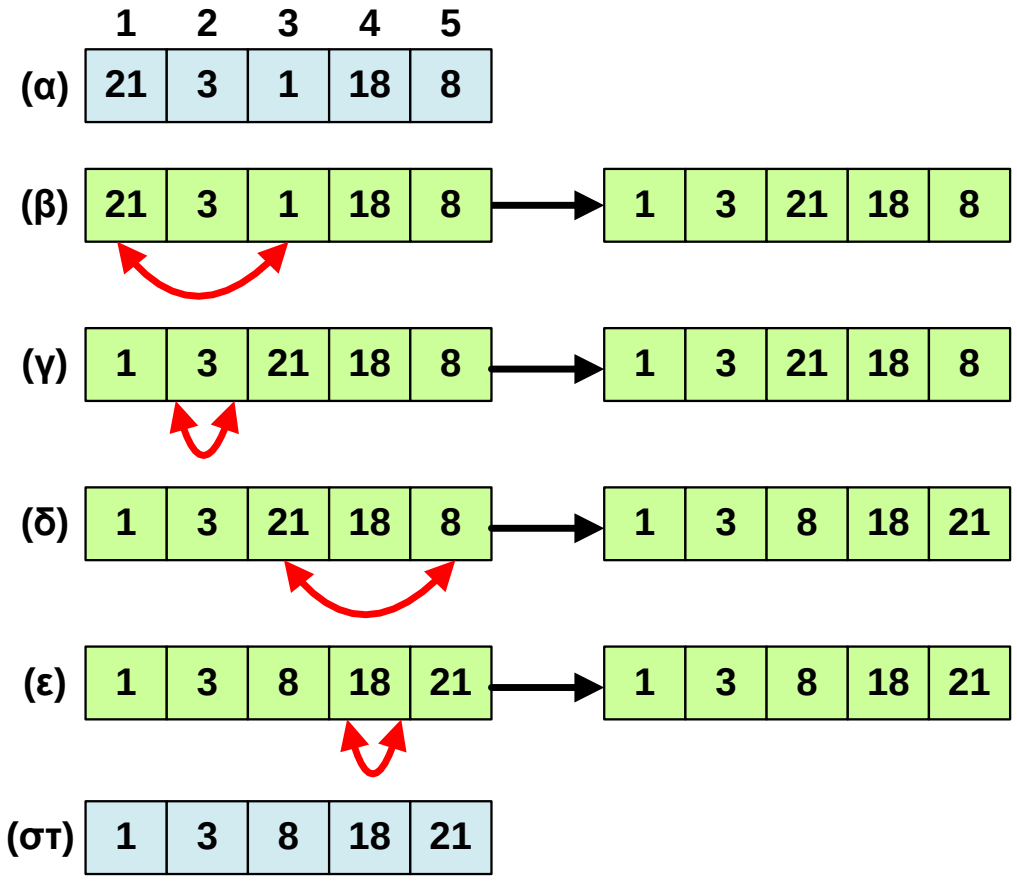


ΑΡΧΗ
Δεδομένα: Πίνακας Α, Ν θέσεων

Διάβασμα αριθμός_προς_αναζήτηση
  Για i=1 έως Ν, βήμα 1
    Αν Α[i]= αριθμός_προς_αναζήτηση ΤΟΤΕ
      Εμφάνιση «Βρέθηκε στη θέση», i
    Τέλος-Αν
  Τέλος-Για
ΤΕΛΟΣ

Πίνακες - Βασικοί υπολογισμοί, Αναζήτηση και ταξινόμηση

Ταξινόμηση με επιλογή

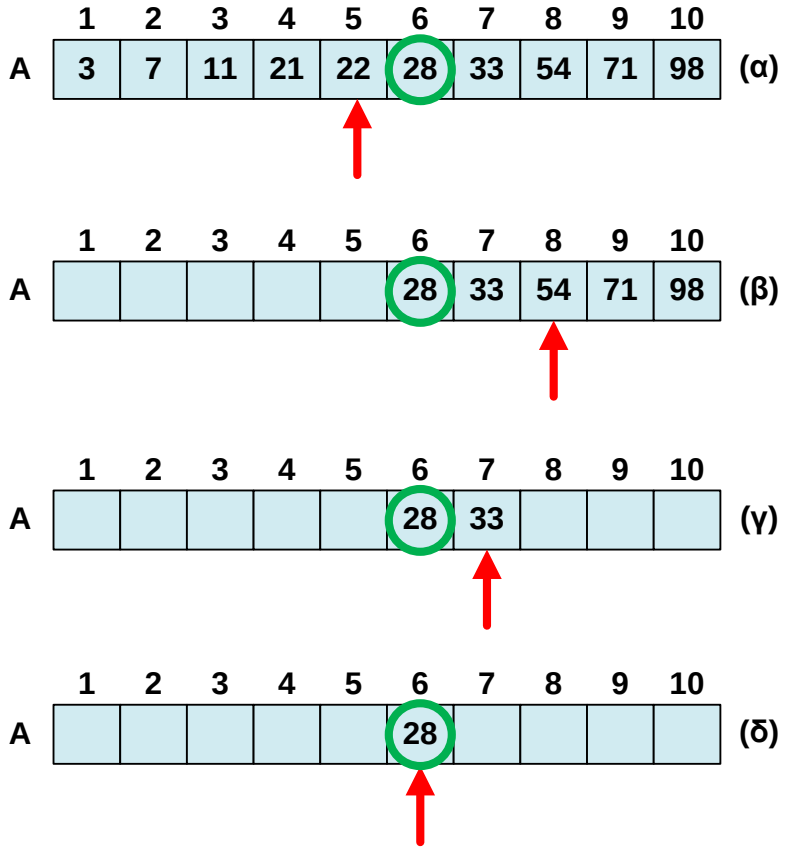


ΑΡΧΗ
Δεδομένα: Πίνακας Α, Ν θέσεων

Για i από 0 έως Ν-1, βήμα 1
  Για j από i+1 έως Ν, βήμα 1
    Αν (A[j]<A[i]) ΤΟΤΕ
    temp=A[j];
  A[j]=A[i];
  A[i]=temp;
ΤΕΛΟΣ

Πίνακες - Βασικοί υπολογισμοί, Αναζήτηση και ταξινόμηση

Δυαδική αναζήτηση



- Εύρεση της μεσαίας θέσης, θεωρώντας start=1 (η πρώτη θέση) και end=10 (η τελευταία θέση)

m=(start + end) / 2 (μόνο το ακέραιο μέρος)

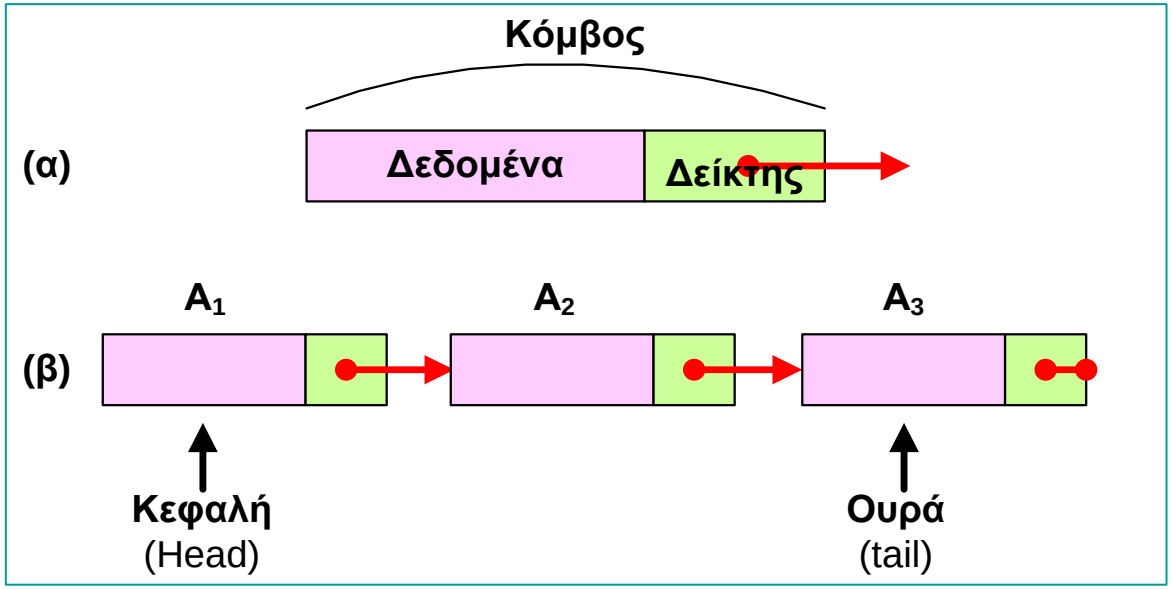
m=(1+10)/2=11/2=5.5 και τελικά m=5

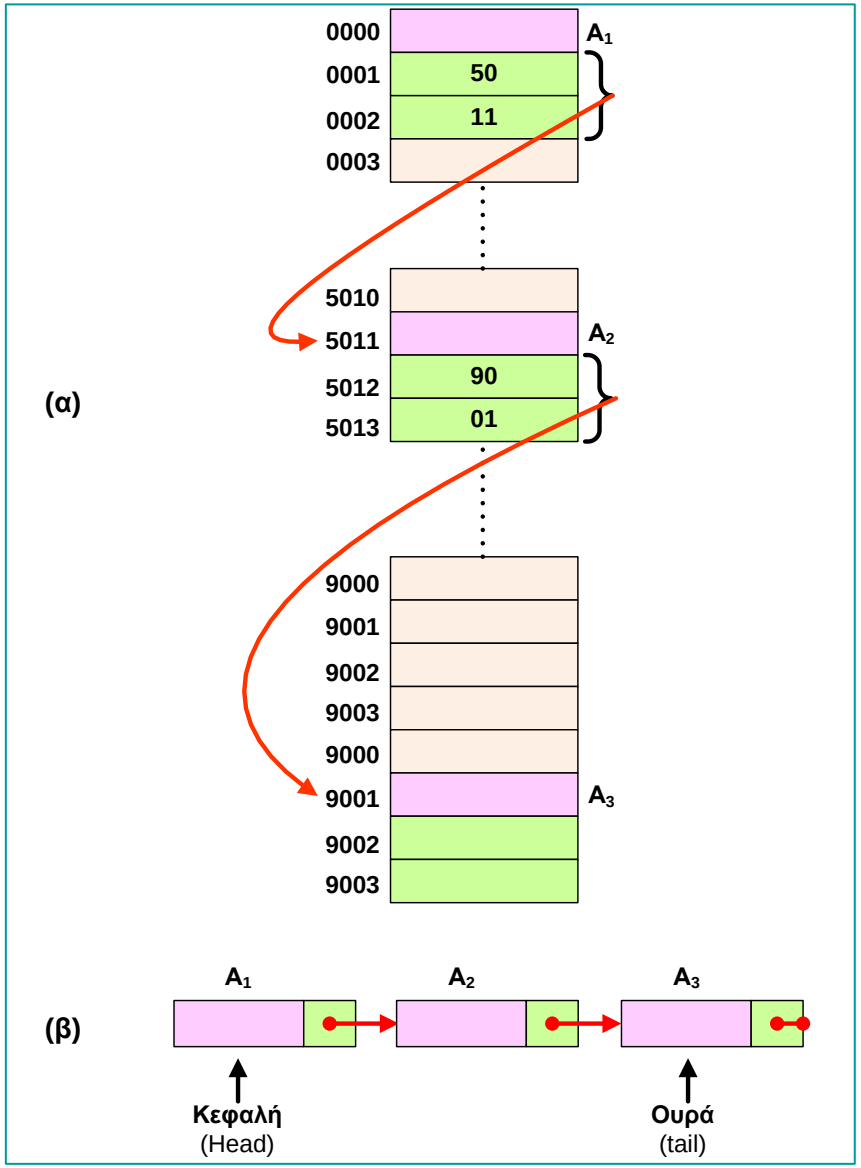
- Αν f>A[m], τότε start=m+1 και συνεχίζεται η αναζήτηση στο δεύτερο μισό
- Αν f<A[m], τότε end=m-1 και η αναζήτηση συνεχίζεται στο πρώτο μισό
- Αν f=A[m], τότε βρέθηκε ο αριθμός και η αναζήτηση τερματίζεται

Διπλανό παράδειγμα

- Αρχικά, f=28 (ο αριθμός που ψάχνουμε), m=5, start=1 και end=16
- Εφόσον f>A[m], δηλαδή 28>22, τότε start=m+1=6 και το νέο τμήμα στο οποίο ψάχνουμε, εκτείνεται στις θέσεις 6 έως 16
- Υπολογίζουμε εκ νέου τη νέα μεσαία θέση m=(start+end)/2=(6+10)/2=8
- Τώρα f<A[m], δηλαδή 28<54 και η αναζήτηση θα συνεχιστεί στο αριστερό μισό του πίνακα που έχει απομείνει και end=m-1=7, δηλαδή στις θέσεις 6 έως 7
- Υπολογίζουμε το νέο μέσο m=(start+end)/2=(6+7)/2=6.5, και τελικά m=6
- Τώρα, ισχύει ότι f=A[m] και ο αριθμός βρέθηκε

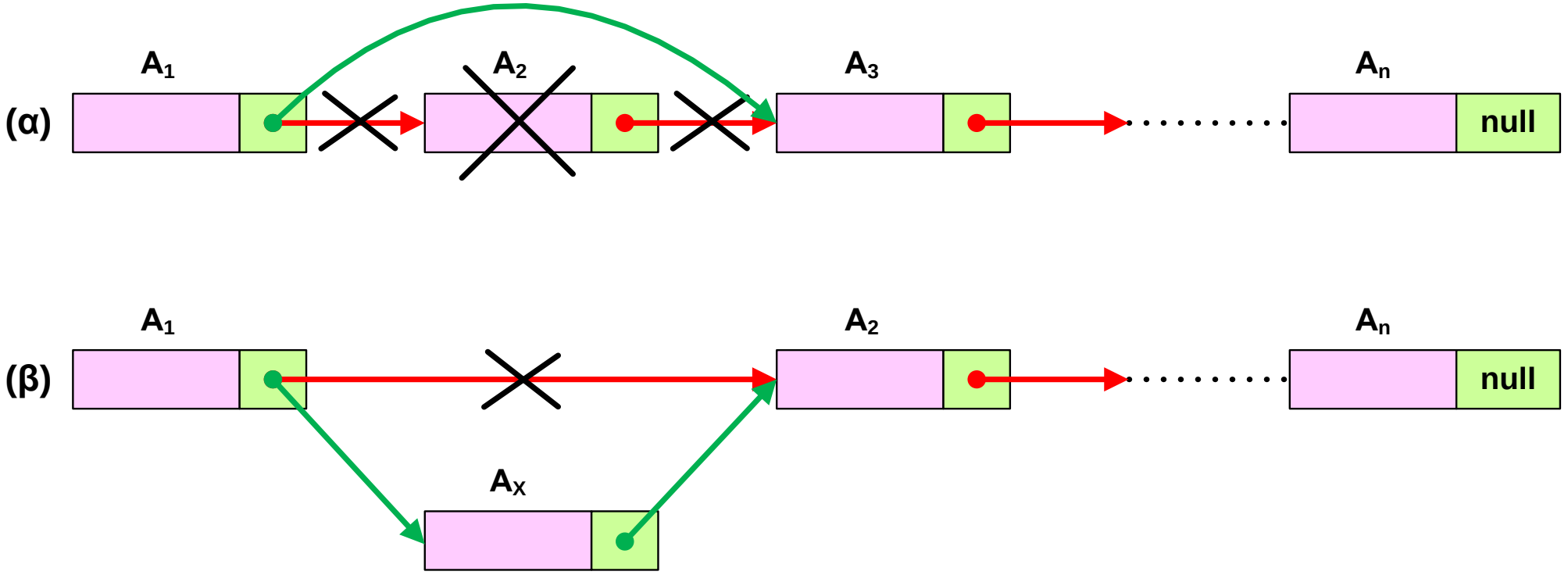
Λίστες
Γενικά





Λίστες

Διαγραφή και εισαγωγή κόμβου



Λίστες
Διαχείριση στο IDLE της Python

Δημιουργία και εμφάνιση λιστών

```
>>> list1=["temp",29.3,"H","80%","bytes",33,2019]
>>> list2=["yellow","red","green"]
>>> list3=["mouse", 12, "keyboard", 17.22]
>>> print(list1)
['temp', 29.3, 'H', '80%', 'bytes', 33, 2019]
>>> print(list2)
['yellow', 'red', 'green']
>>> print(list3)
['mouse', 12, 'keyboard', 17.22]
```

1

Ταξινόμηση λίστας

```
>>> list4=[33, 29, 44, 1, 9, 28]
>>> list2.sort()
>>> list4.sort()
>>> print(list2)
['green', 'red', 'yellow']
>>> print(list4)
[1, 9, 28, 29, 33, 44
```

2

Επιστημονικές Εκδόσεις ΤΖΙΟΛΑ

Λίστες
Διαχείριση στο IDLE της Python

Προσθήκη στοιχείου

```
>>> list2.append("blue")
>>> print(list2)
['green', 'red', 'yellow', 'blue']
>>> list2.sort()
>>> print(list2)
['blue', 'green', 'red', 'yellow']
```

3

Εμφάνιση επιλεγμένων στοιχείων

```
>>> list2[0:2]
['blue', 'green']
>>> list2[2:]
['red', 'yellow']
```

4

Λίστες
Διαχείριση στο IDLE της Python

Διαγραφή στοιχείων

```
>>> del list2[3]
>>> print(list2)
['blue', 'green', 'red']
```

5

Εισαγωγή στοιχείου σε συγκεκριμένη θέση

```
>>> print(list1)
['temp', 29.3, 'H', '80%', 'bytes', 33, 2019]
>>> list1.insert(1,"test")
>>> print(list1)
['temp', 'test', 29.3, 'H', '80%', 'bytes', 33, 2019]
```

6

Επιστημονικές Εκδόσεις ΤΖΙΟΛΑ

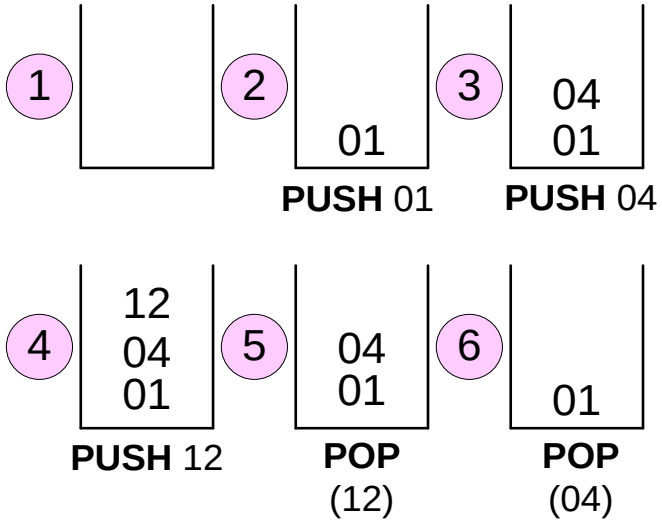
Λίστες
Διαχείριση στο IDLE της Python

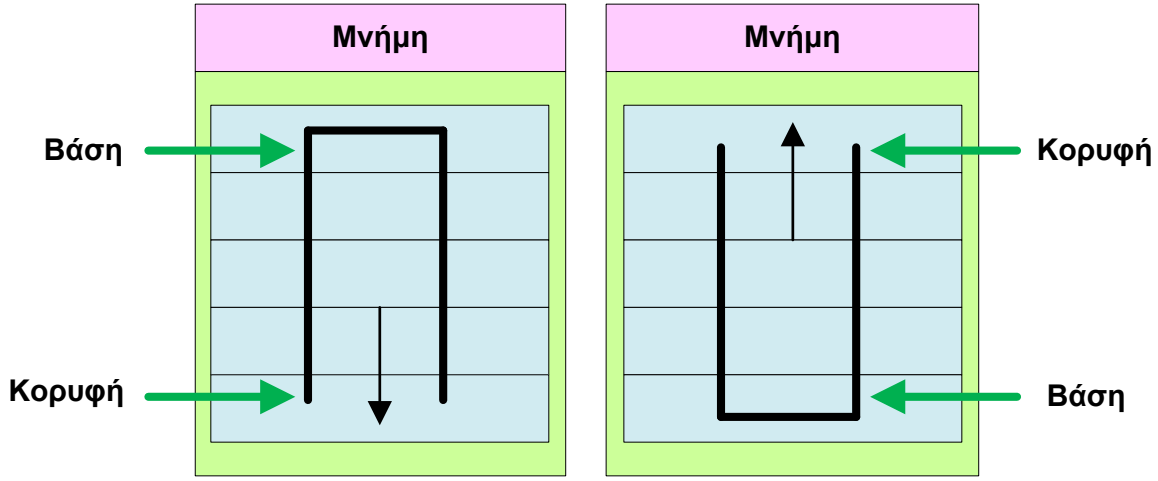
Διαγραφή στοιχείο από συγκεκριμένη θέση

```
>>> print(list1)
['temp', 'test', 29.3, 'H', '80%', 'bytes', 33, 2019]
>>> list1.pop(2)
29.3
>>> print(list1)
```

7

Σωρός (1)





Βήμα 1: Ο σωρός είναι κενός

Βήμα 2: Αποθήκευση (ώθηση) του αριθμού 01 στην κορυφή του σωρού (διαδικασία PUSH)

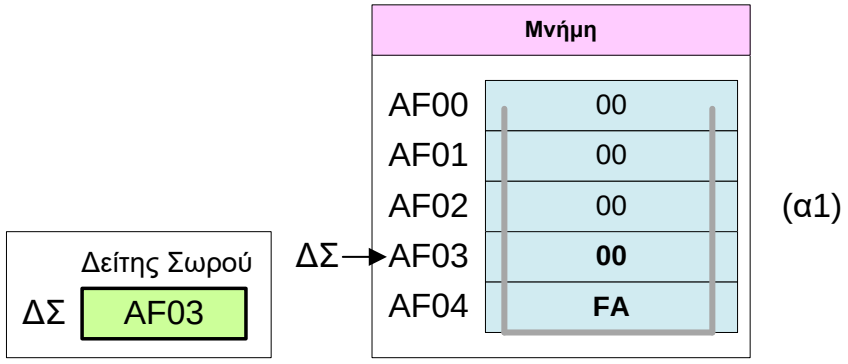
Βήμα 3: Αποθήκευση (ώθηση) του αριθμού 04 στην κορυφή του σωρού (διαδικασία PUSH)

Βήμα 4: Αποθήκευση (ώθηση) του αριθμού 12 στην κορυφή του σωρού (διαδικασία PUSH)

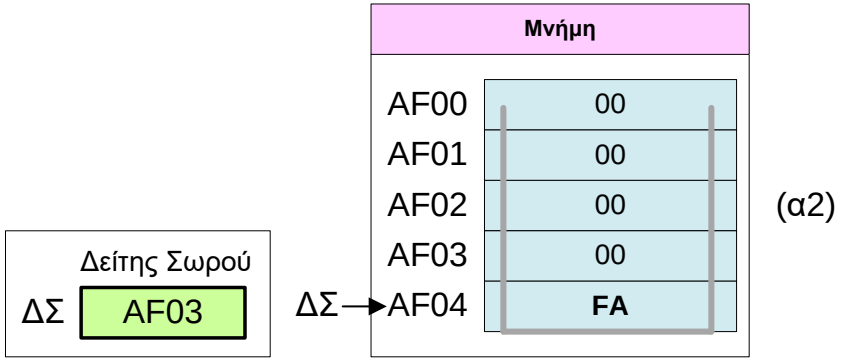
Βήμα 5: Ανάκτηση (απώθηση) του αριθμού από την κορυφή του σωρού (αριθμός 12, διαδικασία POP)

Βήμα 6: Ανάκτηση (απώθηση) του αριθμού από την κορυφή του σωρού (αριθμός 04, διαδικασία POP)

Σωρός (2)
Λειτουργία δείκτη σωρού

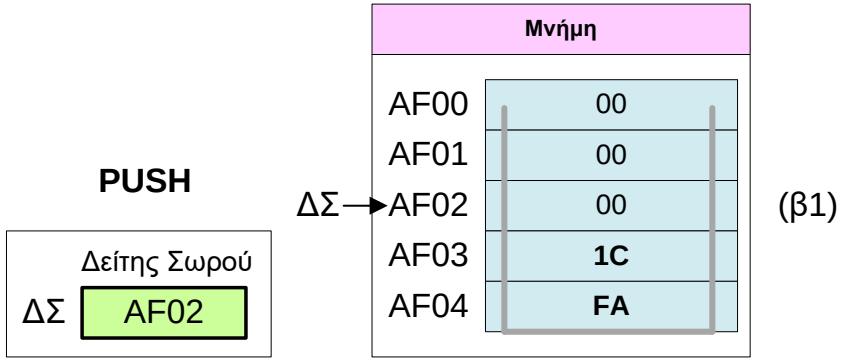


(α1)



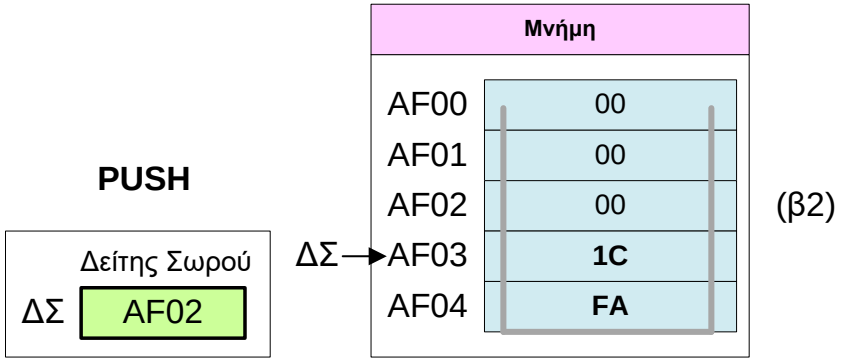
(α2)

PUSH



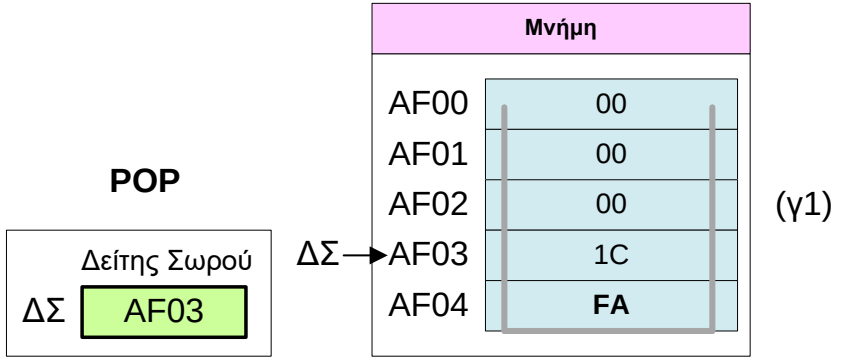
(β1)

PUSH



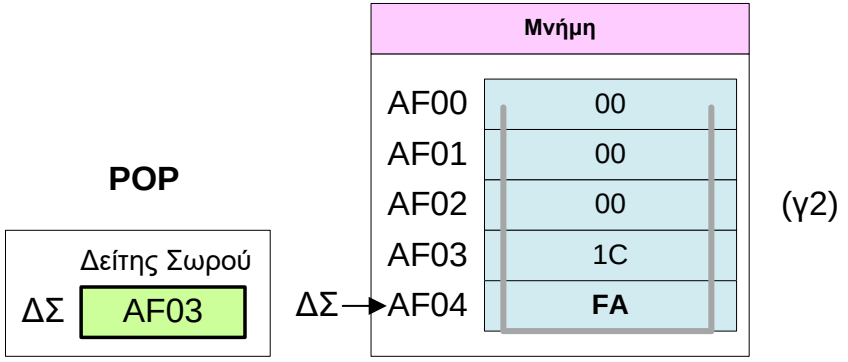
(β2)

POP



(γ1)

POP



(γ2)

Δείτε τις εφαρμογές από το βιβλίο με την υλοποίηση όλων των αλγορίθμων που παρουσιάστηκαν

Επιστημονικές Εκδόσεις
ΤΖΙΟΛΑ

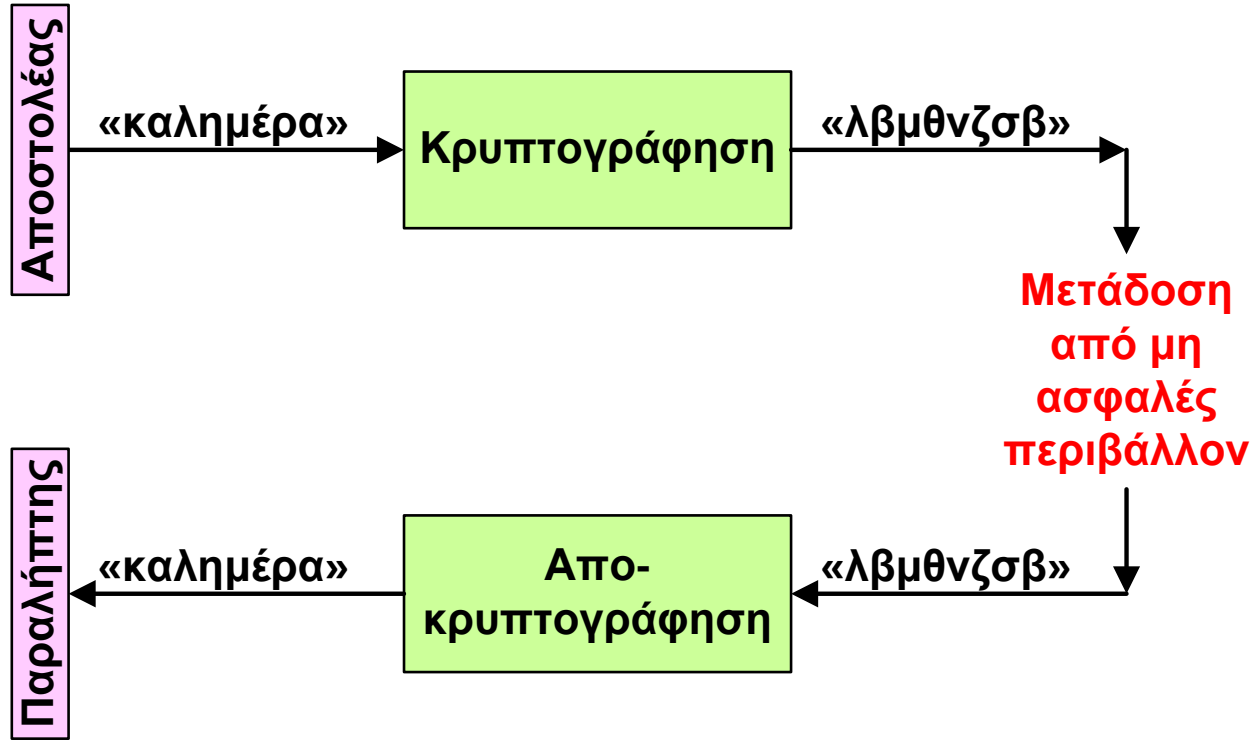
Η ανάγκη κρυπτογράφησης δεδομένων

Ασφάλεια : προστασία δεδομένων και εφαρμογών από την πρόσβαση μη εξουσιοδοτημένων χρηστών

Ένα απλό παράδειγμα (1)

- Θα κρυπτογραφήσουμε το μήνυμα «καλημέρα»
- Μέθοδος: αντικατάσταση κάθε γράμματος με το επόμενό του

Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω



Η ανάγκη κρυπτογράφησης δεδομένων

Ένα απλό παράδειγμα (2)

Κώδικας C

```
#include <stdio.h>

int main()
{
    char text[]={"Finally, this new framework can be used for developing a true computer based dynamic adaptive scaffolding"};
int L=strlen(text);
int i;

char textc[L];
char text0[L];

printf("\n");

for (i=0;i<L;i++)
    textc[i]=text[i]+1;

for (i=0;i<L;i++)
    text0[i]=textc[i]-1;

printf("TEXT = %s\n\n",text);
printf("TEXT-C = %s\n\n",textc);
printf("TEXT-0 = %s\n",text0);

    return 0;
}
```

Η ανάγκη κρυπτογράφησης δεδομένων

Ένα απλό παράδειγμα (3)

TEXT = Finally, this new framework can be used for developing a true computer based dynamic adaptive scaffolding

TEXT-C = Gjobmmz-!uijt!ofx!gsbnfxpsl!dbo!cf!vtfe!gps!efwfmpqjoh!b!usvf!dpnqvufs!cbtfe!ezobnjd!bebqujwf!tdbggpmejoh

TEXT-0 = Finally, this new framework can be used for developing a true computer based dynamic adaptive scaffolding

Ολοκλήρωση κεφαλαίου
Δείτε τις ασκήσεις από το βιβλίο

Επιστημονικές Εκδόσεις ΤΖΙΟΛΑ