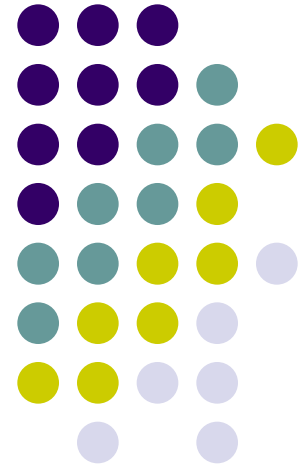
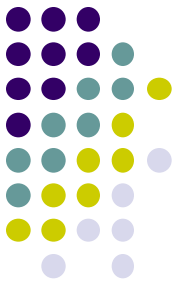
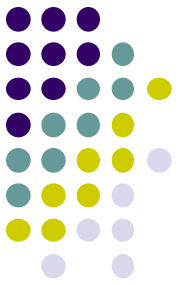


Κλασικοί Αλγόριθμοι II

– Δυαδική Αναζήτηση

Γ' Πληροφορική



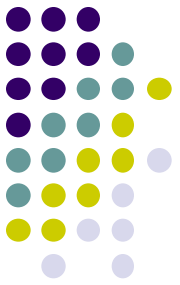


Παραγωγή τυχαίων αριθμών

```
import random  
num=random.randint(1,10)
```

Σημειώσεις

- Η **import random** εισάγει μια βιβλιοθήκη συναρτήσεων για την παραγωγή τυχαίων αριθμών.
- Η **συνάρτηση randint(1,10)** παίρνει ως παραμέτρους ένα κάτω (1) και ένα άνω όριο (10) και επιστρέφει έναν ακέραιο μεταξύ αυτών των ορίων, συμπεριλαμβανομένων και αυτών των δύο δηλαδή, έναν από τους 1, 2, 3, 4, 5, 6, 7, 8, 9, 10



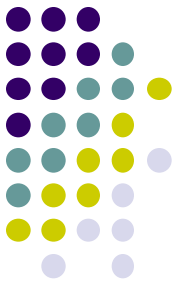
Παραγωγή τυχαίων αριθμών

```
num=random.randrange(1,10)
```

- Η **συνάρτηση randrange(1,10)** επιστρέφει έναν τυχαίο ακέραιο αριθμό μεταξύ 1 και 9, δηλαδή, έναν από τους 1, 2, 3, 4, 5, 6, 7, 8, 9

```
num=random.randrange(10)
```

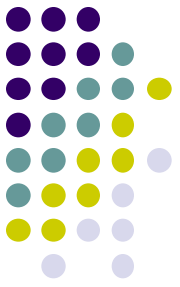
- Η **συνάρτηση randrange(10)** επιστρέφει έναν τυχαίο ακέραιο αριθμό μεταξύ 0 και 9, δηλαδή, έναν από τους 0, 1, 2, 3, 4, 5, 6, 7, 8, 9



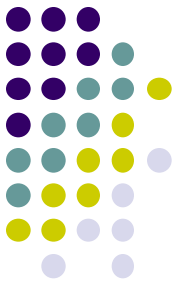
Άσκηση

```
import random
num = random.randint(1,100)
print "Μπορείς να μαντέψεις ένα αριθμό απότο 1 έωςτο 100"
guess = 0
while guess != num:
    guess = input("Δώσε αριθμό: ")
    if guess>num:
print "Έδωσες μεγαλύτερο αριθμό"
    elif guess<num:
print "Έδωσες μικρότερο αριθμό"
    else:
print "Τον βρήκες!"
```

Ένα παιχνίδι

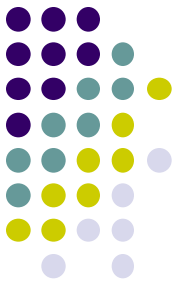


- Σκεφτείτε ένα αριθμό από το 1 έως το 100 και απαντήστε στην ερώτηση:
- Ο αριθμός που σκεφτήκατε είναι μεγαλύτερος ή μικρότερος του 50;
- Ας υποθέσουμε ότι ο αριθμός που σκεφτήκατε είναι μικρότερος του 50.
- Τότε θα βρίσκεται σίγουρα μεταξύ του 1 και του 50.
- Άρα με μια ερώτηση καταφέραμε να περιορίσουμε το χώρο αναζήτησης στο μισό.
- Με μια δεύτερη ερώτηση για τη σχέση του αριθμού μας με το 25 και ανεξάρτητα από την απάντηση, περιορίζουμε το πρόβλημα στο μισό του μισού, δηλ. Στο $\frac{1}{4}$ του αρχικού προβλήματος.

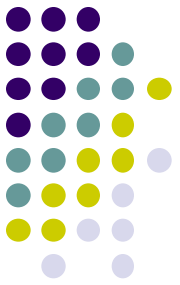


Ένα παιχνίδι (συνέχεια)

- Η προηγούμενη μέθοδος είναι γνωστή ως **Δυαδική αναζήτηση**, και
- εκμεταλεύεται τη διάταξη των στοιχείων ενός συνόλου,
- διαμεράζοντας κάθε φορά το σύνολο σε δύο ίσα μέρη, και
- εφαρμόζοντας πάλι την ίδια μέθοδο στο υποσύνολο στο οποίο ανήκει ο ζητούμενος αριθμός.

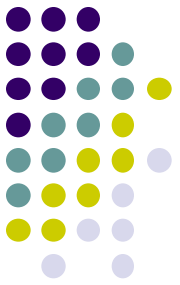


Δυαδική Αναζήτηση (binary search)



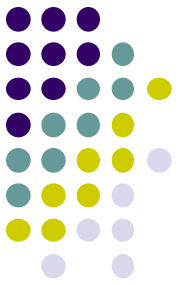
Δυαδική Αναζήτηση

- Η αναζήτηση εφαρμόζεται σε διατεταγμένα σύνολα.
- Με τη δυαδική αναζήτηση ρωτάμε αν το στοιχείο που ψάχνουμε είναι στο μέσο. Η απάντηση που λαμβάνουμε είναι αν το στοιχείο που ψάχνουμε είναι ίσο (άρα το βρήκαμε!), μικρότερο ή μεγαλύτερο από αυτό που ρωτήσαμε.
- Όσο δεν έχουμε βρει το στοιχείο, συνεχίζουμε αποκλείοντας τα μισά στοιχεία και ψάχνοντας κάθε φορά στα υπόλοιπα μισά...



Παράδειγμα

- Θα ακολουθήσει ένα παράδειγμα αναζήτησης του αριθμού 9 μέσα από το διάστημα από το 1 έως το 10.

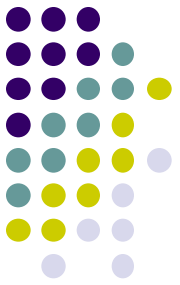


Δυαδική αναζήτηση

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----



- Όταν αρχίζουμε το **“μεσαίο”** στοιχείο είναι το 5. Αυτό προκύπτει από την πράξη $(1+10) / 2$ όπου το 1 είναι το **πρώτο** στοιχείο και το 10 το **τελευταίο**.
- Στο ερώτημα **“είναι το 5;”** η απάντηση που λαμβάνουμε μπορεί να είναι μια από τις:
 - **“είναι”** οπότε το βρήκαμε,
 - **“το 5 είναι μικρότερο”** ή
 - **“το 5 είναι μεγαλύτερο”**.

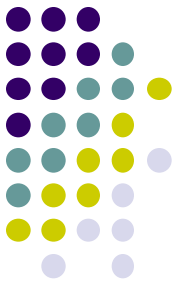


Δυαδική αναζήτηση

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----



- Αφού η απάντηση που λαμβάνουμε είναι **“το 5 είναι μικρότερο”** γνωρίζουμε ότι το στοιχείο που ψάχνουμε είναι στο διάστημα **6 έως 10**.
- Άρα, αποκλείουμε τα στοιχεία από 1 έως και 5,
- και υπολογίζουμε τον νέο μέσο με τον τύπο **$(6+10)/2$**

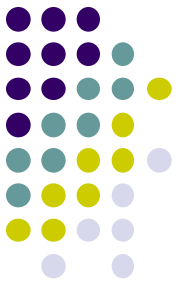


Δυαδική αναζήτηση

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----



- Τώρα θέτουμε ερώτημα ***“είναι το 8;”*** και
- η απάντηση που λαμβάνουμε είναι
- ***“το 8 είναι μικρότερο”***.
- Οπότε, αποκλείουμε και τα 6, 7, 8

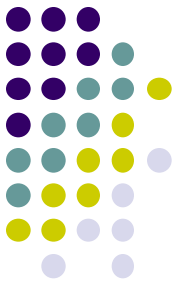


Δυαδική αναζήτηση

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----



- Τώρα θέτουμε ερώτημα **“είναι το 8;”** και η απάντηση που λαμβάνουμε είναι **“το 8 είναι μικρότερο”**.
- Οπότε, αποκλείουμε και τα 6, 7, 8 και
- συνεχίζουμε το ψάξιμο στα στοιχεία που περισσεύουν υπολογίζοντας τον νέο μέσο με τον τύπο **$(9+10)/2$**

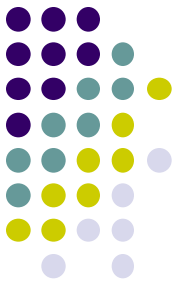


Δυαδική αναζήτηση

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----



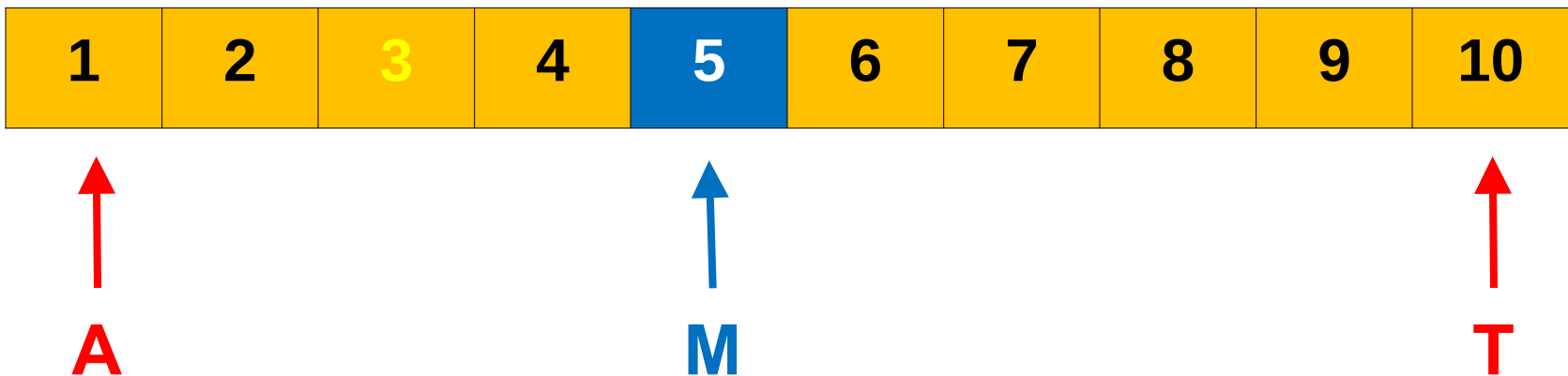
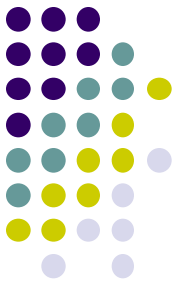
Στο ερώτημα **“είναι το 9;”** η απάντηση είναι
θετική, οπότε τελειώσαμε ...
μετά από τρεις αναζητήσεις !!!



Δυαδική αναζήτηση

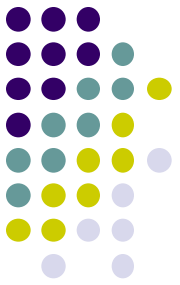
- Ο αλγόριθμος που μόλις περιγράψαμε είναι γνωστός ως **αλγόριθμος δυαδικής αναζήτησης**, γιατί σε κάθε βήμα χωρίζει το χώρο αναζήτησης στο μισό.
- Για να υλοποιηθεί εκτός από τον εκάστοτε **μέσο** χρειάζεται να διαχειριζόμαστε την **αρχή** και το **τέλος** ...
- Έστω ότι ο αριθμός που ψάχνουμε είναι ο **3**

Δυαδική αναζήτηση



Την πρώτη φορά **A=1**, **T=10** και **$M=(A+T)/2$**
Επειδή **$3 < M$** ,

Δυαδική αναζήτηση



1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----



A



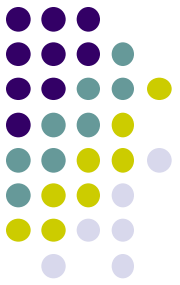
T



M

Την πρώτη φορά **A=1**, **T=10** και **$M=(A+T)/2$**
Επειδή **$3 < M$** , το **T** γίνεται **$T=M-1$**

Δυαδική αναζήτηση



1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----



A



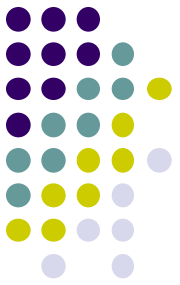
M



T

Την πρώτη φορά **A=1**, **T=10** και **$M=(A+T)/2$**
Επειδή **$3 < M$** , το **T** γίνεται **$T=M-1$** και το
 $M=(A+T)/2$

Δυαδική αναζήτηση



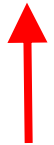
1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----



A



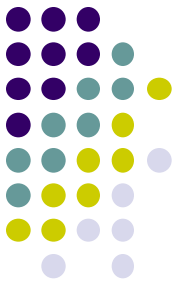
M



T

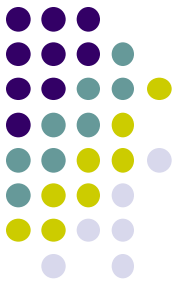
Την δεύτερη φορά **A=1**, **T=4** και **M=2**
Επειδή **3>M**,

Δυαδική αναζήτηση

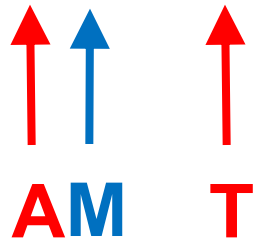


Την δεύτερη φορά **A=1**, **T=4** και **M=2**
Επειδή **3 > M**, το **A** φίνεται **A=M+1**

Δυαδική αναζήτηση

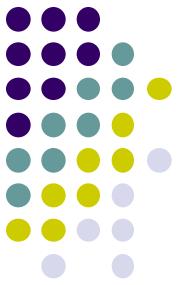


1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

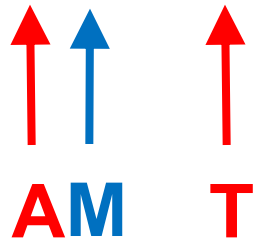


Την δεύτερη φορά **A=1**, **T=4** και **M=2**
Επειδή **3 > M**, το **A** φίνεται **A=M+1**, και το
M=(A+T)/2

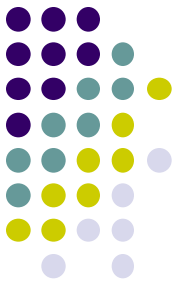
Δυαδική αναζήτηση



1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----



Επειδή $3=M$, η αναζήτηση ολοκληρώθηκε.

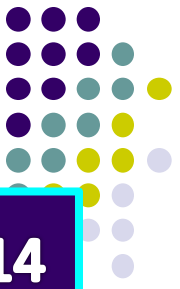


Παράδειγμα

```
import random
number=random.randint(1, 10)
found=False
tr,a,t=0,1,10
while not found:
    tr=tr+1
    m=(a+t)/2
    if m==number:
        found=True
    elif m>number:
        t=m-1
    else: a=m+1
print "Ο αριθμός είναι ο:",m
print "Βρέθηκε μετά από",tr,"προσπάθειες"
```

Δυαδική αναζήτηση σε λίστα

Εκτέλεση του αλγορίθμου δυαδικής αναζήτησης, σε μια λίστα 14 αριθμών, για την αναζήτηση του αριθμού 45



0 1 2 3 4 5 6 7 8 9 10 11 12 13

5 10 17 23 28 30 35 40 45 50 60 63 68 70



45 > 35



0 1 2 3 4 5 6 7 8 9 10 11 12 13

5 10 17 23 28 30 35 40 45 50 60 63 68 70



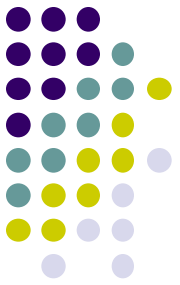
45 < 60



0 1 2 3 4 5 6 7 8 9 10 11 12 13

5 10 17 23 28 30 35 40 45 50 60 63 68 70

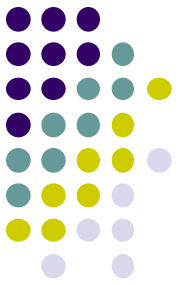




Δυαδική αναζήτηση σε λίστα (με τη βοήθεια συνάρτησης)

```
def binarySearch(array,key):  
    first = 0  
    last = len(array)-1  
    found = False  
    while first <= last and not found:  
        mid = (first + last) / 2  
        if array[mid] == key:  
            found = True  
        elif array[mid] < key:  
            first = mid + 1  
        else:  
            last = mid-1  
    return found  
a = [20,24,34,45,50]  
key=24  
fou=binarySearch(a,key)  
if fou==True:  
    print "vrethike"  
else:  
    print "den vrethike"
```

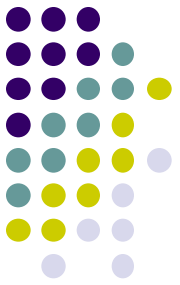
Σημείωση: Ο αλγόριθμος δυαδικής αναζήτησης με τη μορφή συνάρτησης για τα στοιχεία μιας λίστας που είναι ταξινομημένα σε αύξουσα σειρά.



Συνάρτηση Δυαδικής αναζήτησης (επιστρέφει και τη θέση του στοιχείου)

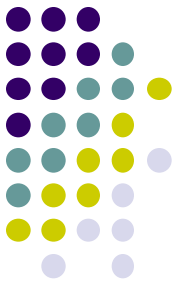
```
def binarySearch(array,key):  
    first = 0  
    last = len(array)-1  
    pos = -1  
    while first <= last and pos==-1:  
        mid = (first + last) / 2  
        if array[mid] == key:  
            pos = mid  
        elif array[mid] < key:  
            first = mid + 1  
        else:  
            last = mid-1  
    return pos
```

Σημείωση: Ο αλγόριθμος επιστρέφει τη θέση του στοιχείου αν υπάρχει, σε διαφορετική περίπτωση επιστρέφει -1.



Πολυμορφισμός

- Οι παραπάνω αλγόριθμοι ισχύουν για όλους τους τύπους δεδομένων για τους οποίους ορίζονται οι συγκριτικοί τελεστές $==$, $<$.
- Δηλαδή μπορούν να χρησιμοποιηθούν για ακεραίους ή πραγματικούς αριθμούς, αλφαριθμητικά ή ακόμα και σύνθετους τύπους για τους οποίους έχουμε ορίσει τους συγκεκριμένους τελεστές.
- Αυτό το χαρακτηριστικό είναι γνωστό ως **πολυμορφισμός**.



Δυναμική αναζήτηση - Σύνοψη

- Εκμεταλεύεται τη διάταξη των στοιχείων ενός συνόλου δεδομένων για τη γρήγορη εύρεση ενός στοιχείου
- Χρησιμοποιείται μόνο σε ταξινομημένες συλλογές δεδομένων
- Βρίσκει το ζητούμενο πολύ πιο γρήγορα από τη σειριακή αναζήτηση

Άσκηση - Δυαδική αναζήτηση



Δίνεται παρακάτω η λίστα **A**.

0 1 2 3 4 5 6 7 8 9 10 11 12 13

A:

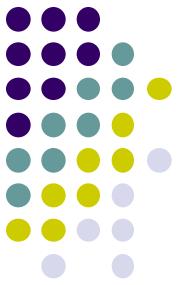
1	1	2	3	5	8	13	21	34	55	89	94	96	99
----------	----------	----------	----------	----------	----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

Να γράψετε στο τετράδιό σας τους αριθμούς της λίστας που θα συγκριθούν κατά την εκτέλεση του αλγορίθμου της δυαδικής αναζήτησης σε κάθε μία από τις παρακάτω περιπτώσεις:

α) για τον αριθμό 100 (μον. 4)

β) για τον αριθμό 1 (μον. 3)

Άσκηση - Δυναδική αναζήτηση (λύση)



0 1 2 3 4 5 6 7 8 9 10 11 12 13

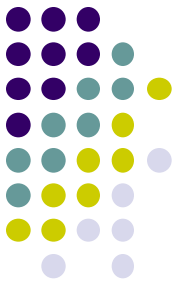
A:

1	1	2	3	5	8	13	21	34	55	89	94	96	99
---	---	---	---	---	---	----	----	----	----	----	----	----	----

α) 13, 89, 96, 99

β) 13, 2, 1

Άσκηση (για λύση) - Δυαδική αναζήτηση



Δίνεται η παρακάτω λίστα L:

L=['ΑΣΗΜΙΝΑ', 'ΓΕΩΡΓΙΑ', 'ΕΛΕΝΗ',
'ΚΩΝΣΤΑΝΤΙΝΟΣ', 'ΜΙΧΑΛΗΣ', 'ΝΑΤΑΛΙΑ', 'ΣΟΦΙΑ']

Να γράψετε πρόγραμμα σε Python το οποίο:

α) να διαβάζει ένα όνομα

β) να αναζητά με τη βοήθεια του αλγορίθμου της δυαδικής αναζήτησης το παραπάνω όνομα στη λίστα L (με συνάρτηση) και

γ) να εμφανίζει σχετικό μήνυμα καθώς και τη θέση του στη λίστα αν υπάρχει