Python – 5. Κλασικοί Αλγόριθμοι II

Δυαδική Αναζήτηση

Σε μορφή συνάρτησης:

```
def binarySearch(L,key):
  first = 0
  last = len(L) - 1
  found = False
  while first <= last and not found:
    mid = (first + last) // 2
    if L[mid] == key:
      found = True
    elif L[mid] < key:
      first = mid + 1
    else:
      last = mid - 1
  return found
```

Παρατηρήστε ότι η συνάρτηση αυτή ισχύει για όλους τους τύπους δεδομένων για τους οποίους ισχύουν οι συγκριτικοί τελεστές ==, <.
Δηλαδή, μπορεί να κληθεί για ακέραιους (int) ή αριθμούς κινητής υποδιαστολής (float), αλφαριθμητικά (str) ή ακόμα και σύνθετους τύπους για τους οποίους έχουμε ορίσει τους συγκεκριμένους τελεστές.
Αυτό το χαρακτηριστικό είναι ένα από τα βασικά πλεονεκτήματα της Python.

Η δυαδική αναζήτηση:

- εκμεταλλεύεται 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 ότι η σειριακή αναζήτηση

Ταξινόμηση Ευθείας Ανταλλαγής ή Ταξινόμηση Φυσαλίδας (Bubble Short)

Σε μορφή συνάρτησης:

```
def BubbleSort(L):
  N = len(L)
  for i in range(1,N,1):
    for j in range(N-1,i-1,-1):
      if L[j] < L[j-1]:
        L[j],L[j-1] = L[j-1],L[j]
  return L
```

Δραστηριότητα 4

Να δώσετε τη **βελτιωμένη** έκδοση του αλγορίθμου ταξινόμησης ευθείας ανταλλαγής η οποία τερματίζει, όταν διαπιστώσει ότι η λίστα είναι ταξινομημένη, ώστε να αποφεύγονται περιττές συγκρίσεις.
Υπόδειξη: Χρησιμοποιήστε μια λογική μεταβλητή η οποία θα αλλάζει τιμή, αν υπάρχουν τουλάχιστον δύο στοιχεία τα οποία δεν βρίσκονται στην επιθυμητή σειρά, καθώς η “φυσαλίδα ανεβαίνει στην επιφάνεια”.
Απάντηση:

```
N=len(L)
sorted=False
i=1
while i<N and not sorted:
  sorted=True
  for j in range(N-1,i-1,-1):
    if L[j] < L[j-1]:
      L[j],L[j-1] = L[j-1],L[j]
      sorted=False
  i=i+1
```

Δραστηριότητα ?

Να μετατρέψετε τον αλγόριθμο φυσαλίδας (στη μορφή με το for) ώστε να κάνει φθίνουσα ταξινόμηση.
Απάντηση:

```
N = len(L)
for i in range(1,N,1):
  for j in range(N-1,i-1,-1):
    if L[j] > L[j-1]:
      L[j],L[j-1] = L[j-1],L[j]
```