Διάλεξη 1: Δομές Δεδομένων και Αλγόριθμοι - Εισαγωγή

Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:

Εισαγωγή στις έννοιες

Δεδομένα και Ενδόμνημη Αναπαράσταση τους

Οργάνωση Δεδομένων και Δομές Δεδομένων

Αλγόριθμοι και Πολυπλοκότητα,

Διδάσκων: Δημήτρης Ζεϊναλιπούρ

Συμβόλαιο Μαθήματος

- **Επίπεδο:** Προπτυχιακό
- **Πίστωση:** 7.0 μονάδες ECTS
- **Προαπαιτούμενα:**
  - ΕΠΛ034: Εισαγωγή στον Προγραμματισμό για ΗΜΜΥ
- **Μέθοδοι Διδασκαλίας**
  - Διαλέξεις (3 ώρες εβδομαδιαίως): Παράδοση Διδασ. Ύλης
  - Φροντιστήριο (1 ώρα εβδομαδιαίως): Ύλη / Θεωρητική Εξάσκηση
  - Εργαστήριο (2 ώρες εβδομαδιαίως): Πρακτική Εξάσκηση

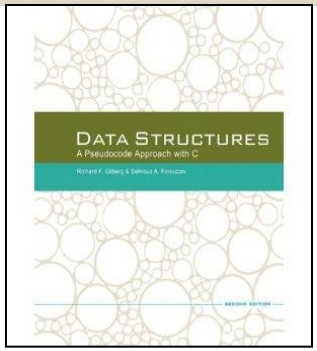
Συμβόλαιο Μαθήματος

- **Αξιολόγηση**
  - 55% Τελική Εξέταση (1)
  - 20% Ενδιάμεση Εξέταση (1)
    - **Ημερ.: Δευτέρα, 24 Οκτωβρίου 2011! (8η Εβδ.)**
  - 25% Ασκήσεις (5 – 6 συνολικά)
    - Προγραμματιστικές, Θεωρητικές ή συνδυασμένες

Βιβλιογραφία

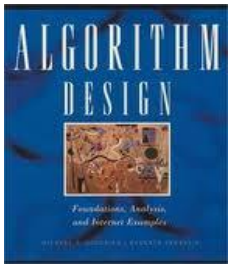
Βασική Βιβλιογραφία

- Data Structures – A Pseudocode Approach with C, Richard F. Gilberg & Behrouz A. Fourouzan, 2nd Edition, Thomson, ISBN: 0-534-39080-3, ISBN-13: 9780534390808, 2005.



Βοηθητική Βιβλιογραφία

- Σημειώσεις Μαθήματος και Συνοδευτικό Υλικό
- Algorithm Design: Foundations, Analysis, and Internet Examples, Michael T. Goodrich and Roberto Tamassia, ISBN-10: 0471383651, ISBN-13: 978-0471383659, 2001.



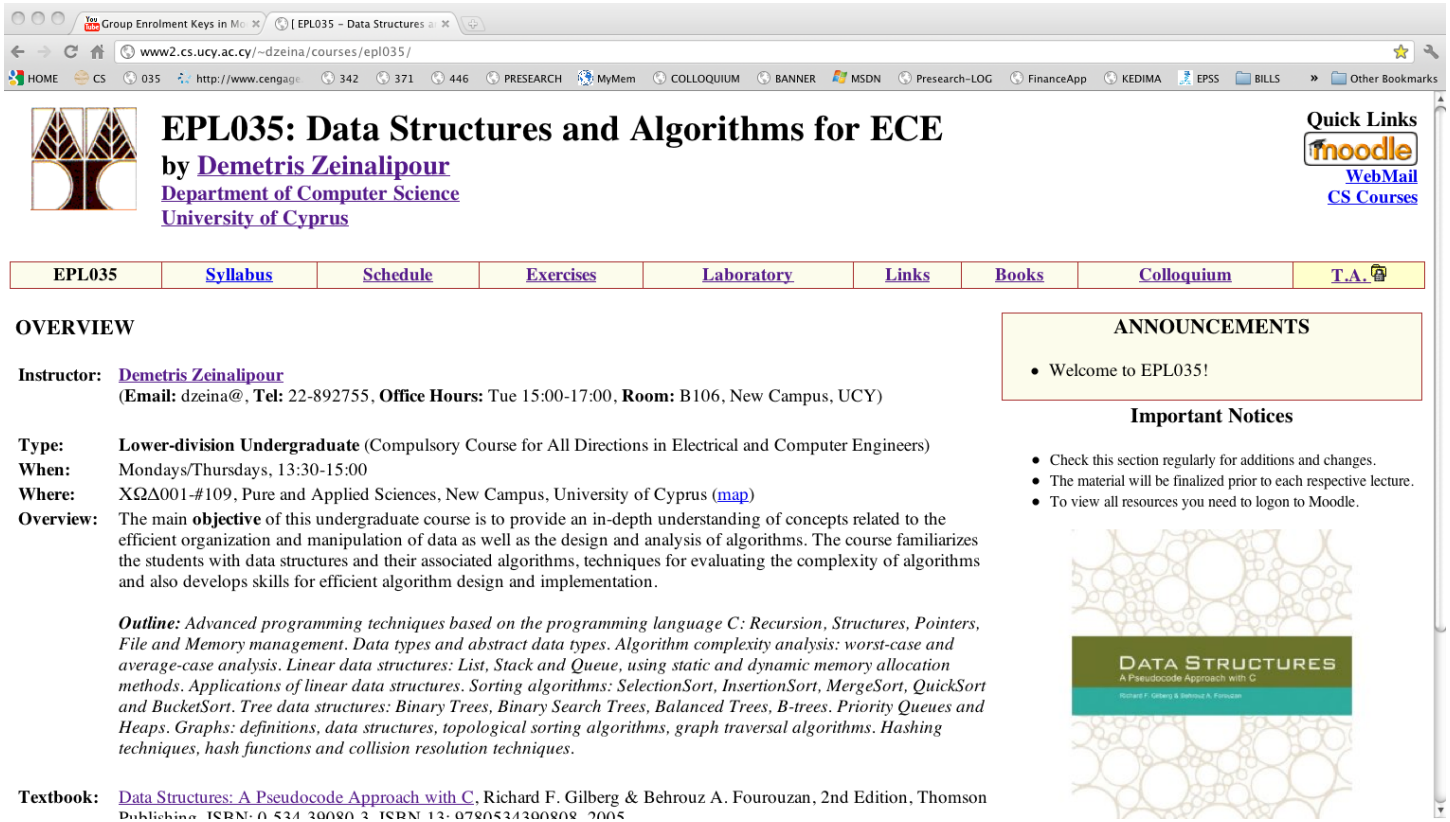
- K.N. King, *C Programming: A Modern Approach*, Second Edition, ISBN-10: 0393979504, ISBN-13: 978-0393979503, 832 pages, W. W. Norton & Company, 2008.

- Νικόλαος Μισυρλής, Δομές Δεδομένων με C, ISBN 960-92031-1-6, 2002.

Ιστοσελίδα ΕΠΛ035

- Όλες οι πληροφορίες σχετικά με το μάθημα βρίσκονται στο ακόλουθο URL

  http://www.cs.ucy.ac.cy/courses/EPL035



Group Enrolment Keys in Mo | [EPL035 – Data Structures

www2.cs.ucy.ac.cy/~dzeina/courses/epl035/

HOME CS 035 http://www.cengage 342 371 446 PRESEARCH MyMem COLLOQUIUM BANNER MSDN Presearch-LOG FinanceApp KEDIMA EPSS BILLS Other Bookmarks

EPL035: Data Structures and Algorithms for ECE

by Demetris Zeinalipour
Department of Computer Science
University of Cyprus

Quick Links
moodle
WebMail
CS Courses

EPL035	Syllabus	Schedule	Exercises	Laboratory	Links	Books	Colloquium	T.A.

OVERVIEW

Instructor: Demetris Zeinalipour
(**Email:** dzeina@, **Tel:** 22-892755, **Office Hours:** Tue 15:00-17:00, **Room:** B106, New Campus, UCY)

Type: **Lower-division Undergraduate** (Compulsory Course for All Directions in Electrical and Computer Engineers)
When: Mondays/Thursdays, 13:30-15:00
Where: ΧΩΔ001-#109, Pure and Applied Sciences, New Campus, University of Cyprus (map)
Overview: The main **objective** of this undergraduate course is to provide an in-depth understanding of concepts related to the efficient organization and manipulation of data as well as the design and analysis of algorithms. The course familiarizes the students with data structures and their associated algorithms, techniques for evaluating the complexity of algorithms and also develops skills for efficient algorithm design and implementation.

Outline: *Advanced programming techniques based on the programming language C: Recursion, Structures, Pointers, File and Memory management. Data types and abstract data types. Algorithm complexity analysis: worst-case and average-case analysis. Linear data structures: List, Stack and Queue, using static and dynamic memory allocation methods. Applications of linear data structures. Sorting algorithms: SelectionSort, InsertionSort, MergeSort, QuickSort and BucketSort. Tree data structures: Binary Trees, Binary Search Trees, Balanced Trees, B-trees. Priority Queues and Heaps. Graphs: definitions, data structures, topological sorting algorithms, graph traversal algorithms. Hashing techniques, hash functions and collision resolution techniques.*

Textbook: Data Structures: A Pseudocode Approach with C, Richard F. Gilberg & Behrouz A. Fourouzan, 2nd Edition, Thomson Publishing, ISBN: 0-534-39080-3, ISBN-13: 9780534390808, 2005

ANNOUNCEMENTS

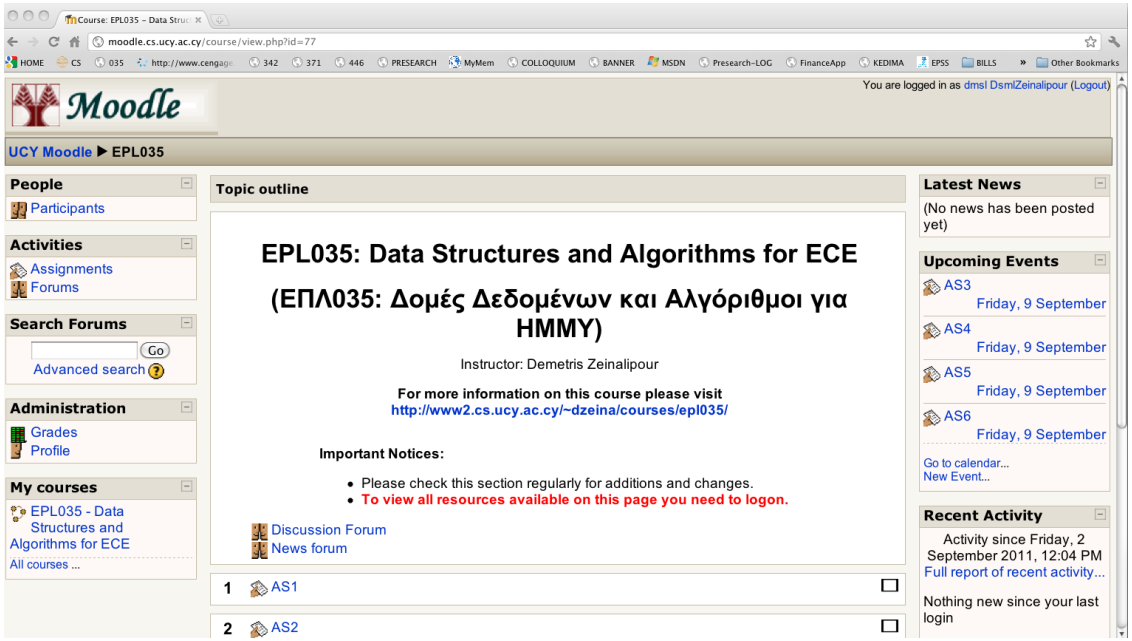
- Welcome to EPL035!

Important Notices

- Check this section regularly for additions and changes.
- The material will be finalized prior to each respective lecture.
- To view all resources you need to logon to Moodle.

ΕΠΛ035 Moodle

- Για τις εκπαιδευτικές δραστηριότητες του μαθήματος (**υποβολή εργασιών**, **φόρουμ** ανακοινώσεων, κτλ) θα χρησιμοποιηθεί το Moodle: http://moodle.cs.ucy.ac.cy/



Εγγραφείτε σήμερα κάνοντας χρήση του Κλειδιού Εγγραφής που θα δοθεί στην τάξη!

Δεδομένα

- **Τι είναι τα *Δεδομένα*;**

  Αφαιρετική όψη της πραγματικότητας.

  Από τα (επεξεργασμένα) δεδομένα προκύπτει η πληροφορία.

- Τα δεδομένα που χρησιμοποιούμε **δεν έχουν πάντοτε τον ίδιο βαθμό πολυπλοκότητας**, π.χ., "Στοιχεία Υπαλλήλου" vs. "Ηλικία"
- Μερικά μπορούν να αναλυθούν σε απλούστερα συστατικά, ενώ άλλα δεν μπορούν. Δεδομένα που δεν μπορούν να αναλυθούν λέγονται ***πρωτόγονα δεδομένα.* – primitive data types (π.χ. int, char)**

- Τα δεδομένα ενός προβλήματος συνήθως δεν είναι μια άμορφη συλλογή στοιχείων. Τις περισσότερες φορές μπορούν να εκφραστούν με γνωστές μαθηματικές δομές, π.χ.
  - σαν ένα απλό σύνολο (διακριτών στοιχείων): {2, 3, 5, 7, 11, 13}
  - σαν ένα διάνυσμα (διατεταγμένων στοιχείων): (2, 3, 5, 7, 11, 13)
  - σαν ένας πίνακας : (2D vector) $\begin{pmatrix} 2 & 3 & 5 \\ 7 & 11 & 13 \end{pmatrix}$

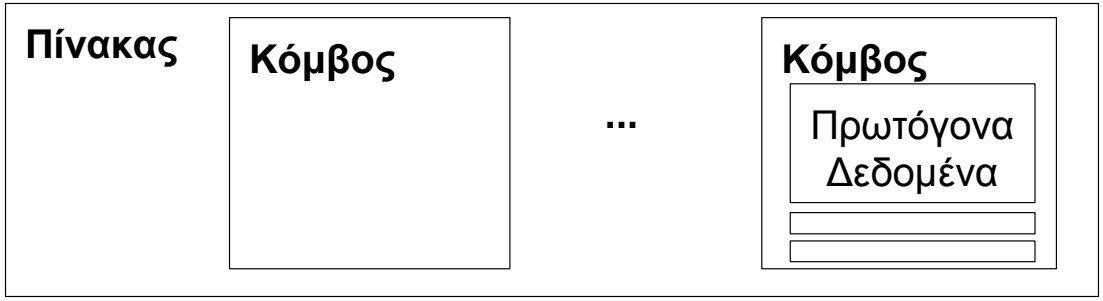
Επομένως χρειαζόμαστε τις κατάλληλες δομές για να τα οργανώσουμε στην μνήμη.

Δεδομένα

- Οι ***δομές δεδομένων*** είναι συλλογές πρωτόγονων δεδομένων, που συνδυάζονται για να σχηματίσουν πολυπλοκότερα δεδομένα.

- Θα χρησιμοποιήσουμε τον όρο ***πίνακας*** για να περιγράψουμε μια συλλογή στοιχείων τα οποία θα ονομάσουμε ***κόμβους*** ή ***καταχωρήματα***.

- Ένας κόμβος μπορεί να είναι ***απλός***, δηλαδή να αποτελείται από ένα μόνο πρωτόγονο δεδομένο, ή να είναι ***σύνθετος***, οπότε αποτελείται από δύο ή περισσότερα ***πεδία***. Τα πεδία ενός κόμβου μπορεί να αντιπροσωπεύουν πρωτόγονα δεδομένα διαφόρων τύπων.




```
struct node{
   char sex; // 8 bits
   int age; // 32 bits
   char owns_car:1; // 1 bit
   int idcard; // 32 bits
 };
```

32/64-bit Αρχιτεκτονικές

- Λίγα λόγια για τύπους δεδομένων με x86 και x64 αρχιτεκτονικές
  - **I**(ntegers) L(ong) P(ointer) 32 => x86 Model
  - L(ong) P(ointer) 64 => x64 Model

Datatype	**ILP32 Model**	**LP64 Model**
char	8	8
short	16	16
int	32	32
long	**32 (4 bytes)**	**64 (8 bytes)**
pointer	**32 (4 bytes)**	**64 (8 bytes)**

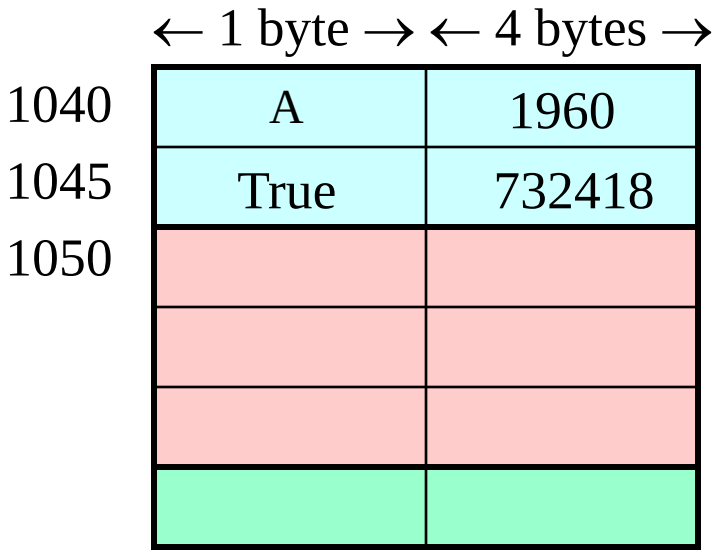
Η μνήμη μπορεί να έχει MONO μέχρι 2^32 = ~4x10^9 (δηλ., 4GB) διευθύνσεις ☹!

Η μνήμη μπορεί να έχει μέχρι 16 Exa (x10^{18}) διευθύνσεις☺!

Δοκιμάστε printf("%ld", sizeof(long));στο υπολογιστή σας!

Δεδομένα

- Ένα πεδίο ή ένα καταχώρημα χαρακτηρίζεται από τη ***διεύθυνση*** του και το ***μήκος*** του. Η διεύθυνση ενός πεδίου ή ενός καταχωρήματος είναι η διεύθυνση της πρώτης του κυψελίδας μνήμης και το μήκος του είναι ο αριθμός κυψελίδων από τις οποίες αποτελείται.




```
#include <stdio.h>

int main() {
        char letter;

        letter = 'a' ;
        printf("%c => %d",a, &a);
}
prints "a => 2289431"
```

address

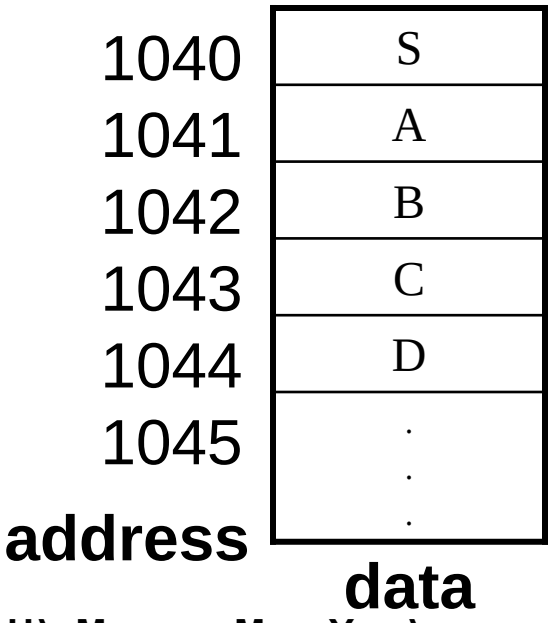
- Η διεύθυνση του πρώτου πεδίου, του πρώτου καταχωρήματος και ολόκληρου του πίνακα είναι 1040. Η διεύθυνση του δεύτερου καταχωρήματος 1050, και ούτω καθεξής.

Ενδόμνημη Παράσταση Δομών Δεδομένων

- Για να αναφερθούμε σε ένα καταχώρημα ή σε ένα πεδίο, χρησιμοποιούμε συνήθως συμβολικά ονόματα (π.χ., array).
- Για να πραγματοποιηθεί μια δομή δεδομένων, απαιτείται η **συσχέτιση** των **συμβολικών ονομάτων** με αντίστοιχα τμήματα της μνήμης του υπολογιστή. Αυτό μπορεί να γίνει με δύο τρόπους:
  - ***Διαδοχική (Στατική) Χορήγηση Μνήμης***

    Παράδειγμα:

    char array[]={ 'S' , 'A' ,' B' ,' C' ,D' };



Ενδόμνημη Παράσταση Δομών Δεδομένων

- **Συνδετική (Δυναμική) Χορήγηση Μνήμης**

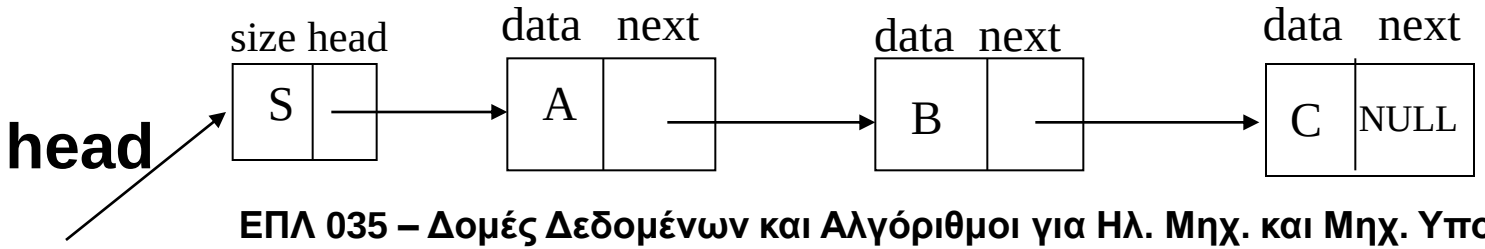
Ένα πεδίο μπορεί να παριστά τη διεύθυνση ενός άλλου πεδίου. Έτσι κατά τη **συνδετική χορήγηση** μνήμης κάθε κόμβος περιέχει πληροφορία σχετικά με το που **βρίσκεται ο επόμενος κόμβος** της δομής.

Για την αποθήκευση αυτής της πληροφορίας κάθε κόμβος χρειάζεται ένα **ειδικό πεδίο** (**pointer**). Το πεδίο αυτό είναι χαρακτηριστικό της συνδετικής χορήγησης μνήμης.

Pointer in C:

NODE *head;

Παράδειγμα: **Συνδεδεμένη Λίστα (linked-list)**



Παράδειγμα: Συνδεδεμένη Λίστα

- Οι κόμβοι της λίστας αποτελούνται από δύο πεδία: Info, για καταχώρηση πληροφοριών (των στοιχείων της λίστας), και Next (τύπου pointer) για αποθήκευση του δείκτη που δείχνει τη διεύθυνση του επόμενου κόμβου της λίστας.

	Info	Next
1040	B	1060
1045		
1050	A	1040
1065		
1060	C	NULL
.	.	.
.	.	.
.	.	.
2030	S	1050

Αλγόριθμοι

Τι είναι ένας αλγόριθμος;

Abu Jafar Mohammed ibn Musa Al-Khowarizmi (790-840)

Αλγόριθμος είναι μια πεπερασμένη ακολουθία εντολών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο (μετρήσιμο) χρόνο, οι οποίες αν ακολουθηθούν επιτυγχάνεται κάποιο επιθυμητό αποτέλεσμα.

Απαραίτητα κριτήρια:

- Υπάρχει είσοδος και έξοδος
- Καθορισμός εντολών (όχι ασάφειες).
- Περατότητα (να διεκπεραιώνει τον στόχο).

Επίδοση: δεν είναι απαραίτητη αλλά επιθυμητή

Δηλαδή, δοθέντος ενός προβλήματος, ένας αλγόριθμος παρέχει τις **οδηγίες** σύμφωνα με τις οποίες τα ***δεδομένα*** του προβλήματος ***μετασχηματίζονται*** και ***συνδυάζονται*** για να προκύψει η **λύση του προβλήματος**.

Αλγόριθμοι

- Κατά την εκτέλεση ενός αλγόριθμου η δομή που έχουν τα δεδομένα παίζει πολύ μεγάλο ρόλο.

- Εξίσωση Wirth

  Αλγόριθμοι + Δεδομένα = Προγράμματα

Κεντρικός στόχος του μαθήματος είναι η μελέτη δομών δεδομένων, αναπαράστασής τους στη μνήμη ενός υπολογιστή, και αλγόριθμων που τις δημιουργούν και τις επεξεργάζονται.

Το πρόβλημα επιλογής (selection problem)

- **Πρόβλημα: Έστω ότι έχουμε n αριθμούς και θέλουμε να προσδιορίσουμε τον k-οστό πιο μεγάλο.**

  π.χ. Έστω οι αριθμοί {5, 72, 3, 4, 1, 9, 65} (72,65,9,5,4,3,1)

  Ο 2^{os} πιο μεγάλος είναι ο 65, ο 6^{os} πιο μεγάλος είναι ο 3, κοκ.

- Αυτό το πρόβλημα είναι γνωστό ως το **πρόβλημα επιλογής (the selection problem).** Υπάρχουν διάφοροι 'εύκολοι' τρόποι λύσης:
  1. **Sort-Based:** 'Διαβάζουμε' τους n αριθμούς σε μια λίστα, a. Ταξινομούμε τη λίστα από το μεγαλύτερο στο μικρότερο με βάση κάποιο αλγόριθμο ταξινόμησης. Επιστρέφουμε το (k-1)οστό στοιχείο της λίστας, δηλαδή το a[k-1].
  2. **Buffer-based:** 'Διαβάζουμε' τους k πρώτους αριθμούς σε μια λίστα a. Ταξινομούμε τη λίστα από το μεγαλύτερο στο μικρότερο δηλ για k=3 έχουμε a={72,5,3}. Μετά, επεξεργαζόμαστε τους υπόλοιπους n-k αριθμούς ως εξής: αν ένα στοιχείο είναι πιο μικρό από το a[k-1] το αγνοούμε, διαφορετικά το τοποθετούμε στη σωστή θέση της λίστας. Όταν η διαδικασία αυτή τελειώσει, επιστρέφουμε το k-οστό στοιχείο της λίστας, δηλαδή το a[k-1].
- Το μεθοδολογία για αποτίμησης της επίδοσης των επί μέρους λύσεων θα είναι αντικείμενο μελέτης αυτού του μαθήματος.

Αλγόριθμοι

- Σημαντικό συμπέρασμα είναι πως το να γράφουμε ένα **σωστό πρόγραμμα δεν είναι αρκετό**. Ιδιαίτερα, όταν το σύνολο αρχικών δεδομένων είναι μεγάλου μεγέθους, ο ***χρόνος εκτέλεσης*** ενός προγράμματος είναι κύριας σημασίας.

- Στο μάθημα αυτό θα μάθουμε να i) **υπολογίζουμε το χρόνο εκτέλεσης αλγόριθμων** και ii) να συγκρίνουμε την αποδοτικότητα διαφορετικών αλγόριθμων, **προτού τους υλοποιήσουμε.** Θα μελετήσουμε επίσης μεθόδους βελτίωσης της ταχύτητας προγραμμάτων.