

Δομές Δεδομένων και Τεχνικές Προγραμματισμού

Μανόλης Κουμπάρκης

Εισαγωγή

- Ιστοσελίδα του μαθήματος: <http://cgi.di.uoa.gr/~k08/>
- Βαθμολόγηση:
 - 5 εργαστήρια: 1.2 μονάδες
 - 3 εργασίες: $3 \times 1.6 = 4.8$ μονάδες
 - Τελική εξέταση: 4 μονάδες
- Περιβάλλον για συζήτηση, απορίες κλπ.: piazza
- Εγγραφή στα εργαστήρια από το e-class.
- Εγγραφή στο piazza από την ιστοσελίδα του μαθήματος.

Γιατί Πρέπει να μας Ενδιαφέρουν οι Δομές Δεδομένων

- Είναι θεμελιώδες τμήμα της Πληροφορικής!
- Τις συναντάμε παντού!
- Ας δούμε μερικά μοντέρνα παραδείγματα ...

Διαδίκτυο, Παγκόσμιος Ιστός: Γράφοι



Μηχανές Αναζήτησης: Inverted Indices

Full inverted index (multiple documents, word-level)



Vocabulary	n_i
to	2
do	3
is	1
be	4
or	1
not	1
I	2
am	2
what	1
think	1
therefore	1
da	1
let	1
it	1

Occurrences as full inverted lists

[1,4,[1,4,6,9]], [2,2,[1,5]]

[1,2,[2,10]], [3,3,[6,8,10]], [4,3,[1,2,3]]

[1,2,[3,8]]

[1,2,[5,7]], [2,2,[2,6]], [3,2,[7,9]], [4,2,[9,12]]

[2,1,[3]]

[2,1,[4]]

[2,2,[7,10]], [3,2,[1,4]]

[2,2,[8,11]], [3,1,[5]]

[2,1,[9]]

[3,1,[2]]

[3,1,[3]]

[4,3,[4,5,6]]

[4,2,[7,10]]

[4,2,[8,11]]

To do is to be.
To be is to do.

d_1

To be or not to be.
I am what I am.

d_2

I think therefore I am.
Do be do be do.

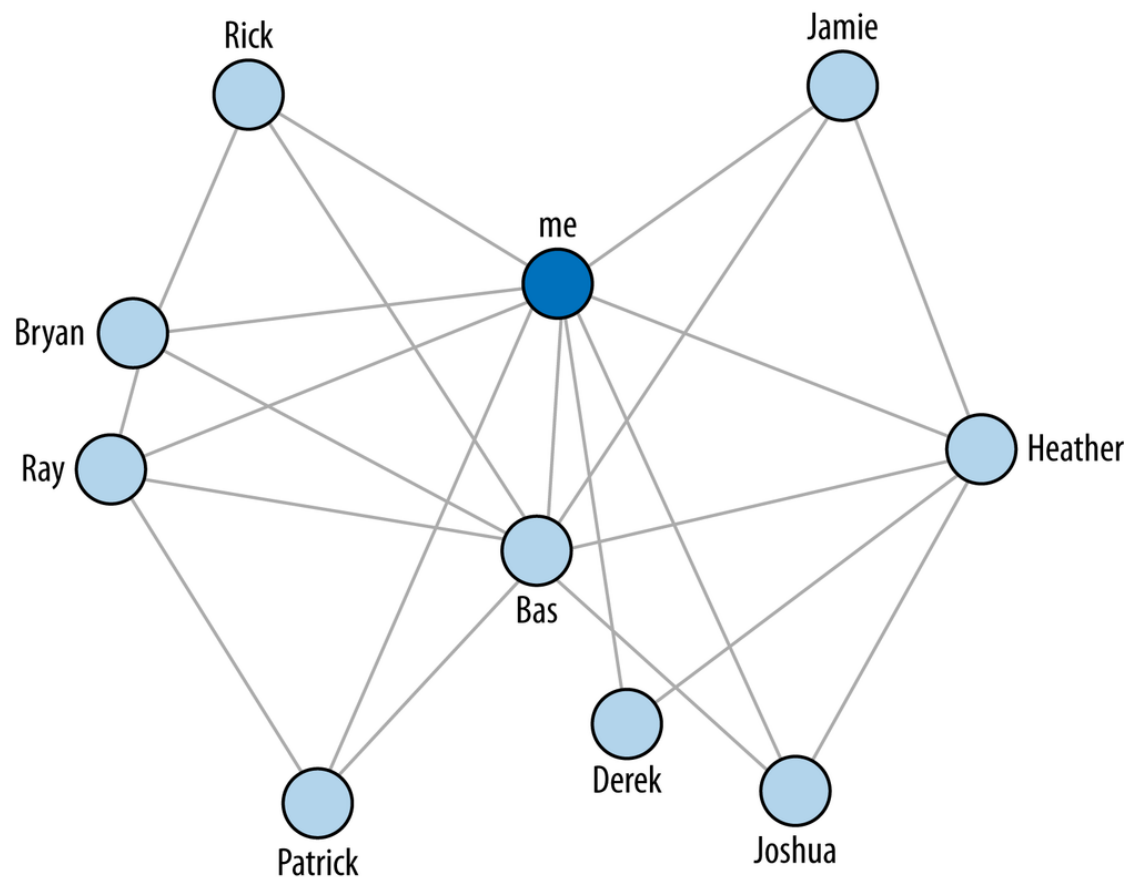
d_3

Do do do, da da da.
Let it be, let it be.

d_4

Κοινωνικά Δίκτυα: Γράφοι

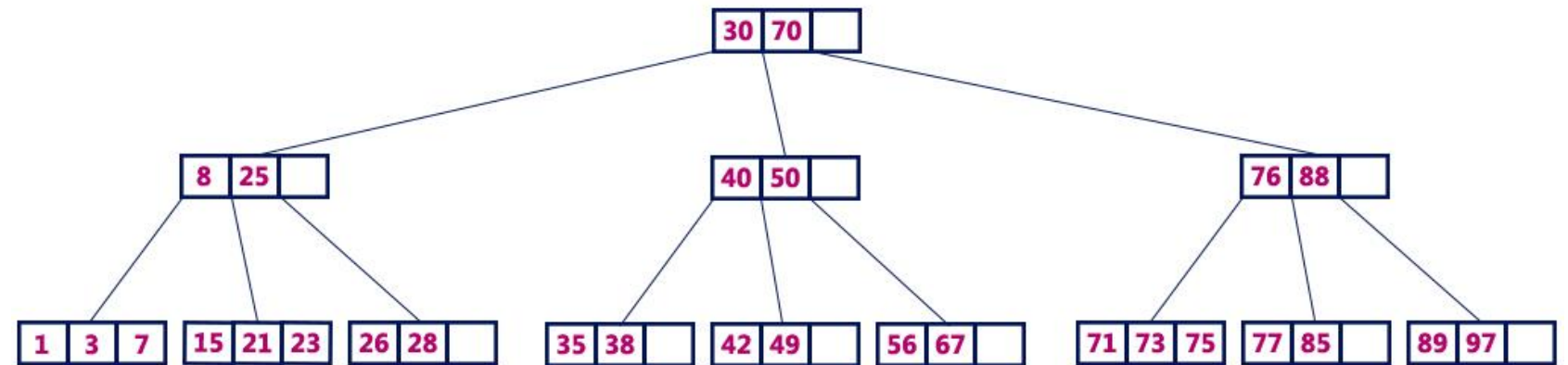
facebook



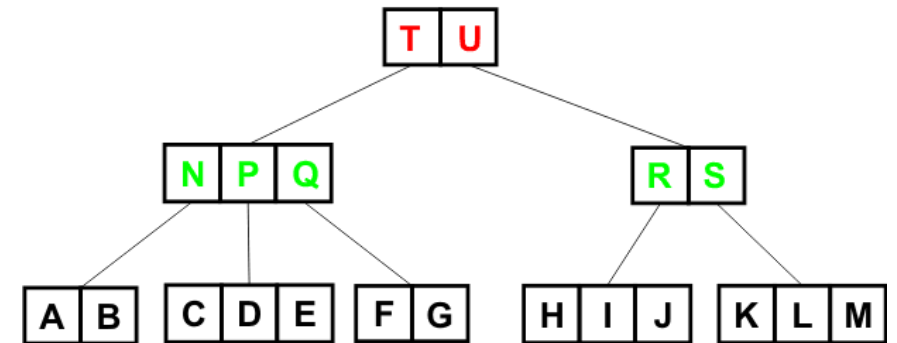
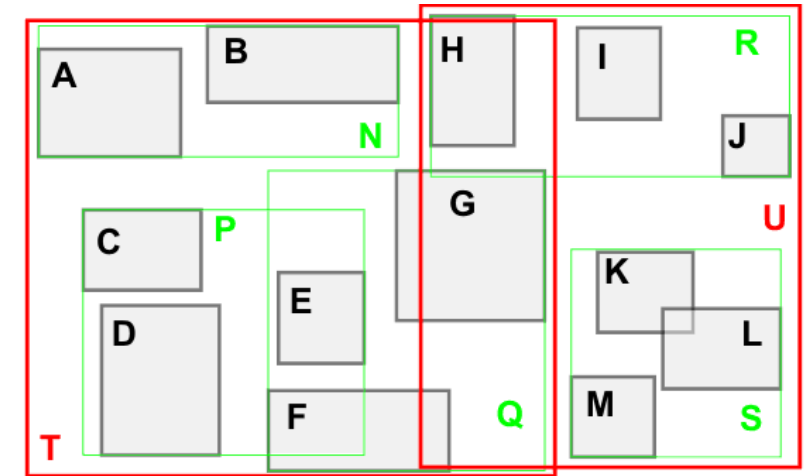
Βάσεις Δεδομένων: Β-δένδρα



B-Tree of Order 4



Εικόνες, Χάρτες: R-Trees



Torrents: Κατανεμημένοι Πίνακες Κατακερματισμού (DHTs)

qBittorrent v4.0.3

File Edit View Tools Help

Filter tor...

STATUS

- All (1)
- Downloading (1)
- Seeding (0)
- Completed (0)
- Resumed (1)
- Paused (0)
- Active (0)
- Inactive (1)
- Errored (0)

CATEGORIES

- All (1)
- Uncategorized (1)

TAGS

- All (1)
- Untagged (1)

TRACKERS

- All (1)
- Trackerless (0)
- Error (0)
- Warning (0)
- coppersurfer.tk (1)
- demonii.com (1)
- desync.com (1)
- leechers-paradise.org (1)
- zer0day.ch (1)

#	Name	Size	Done	Status	Seeds	Peers	Down Speed	Up Speed	ETA	Ratio
1	Dunkirk (201...	1.64 GiB	56.2%	Checking	0 (0)	0 (0)	0 B/s	0 B/s	∞	0.00

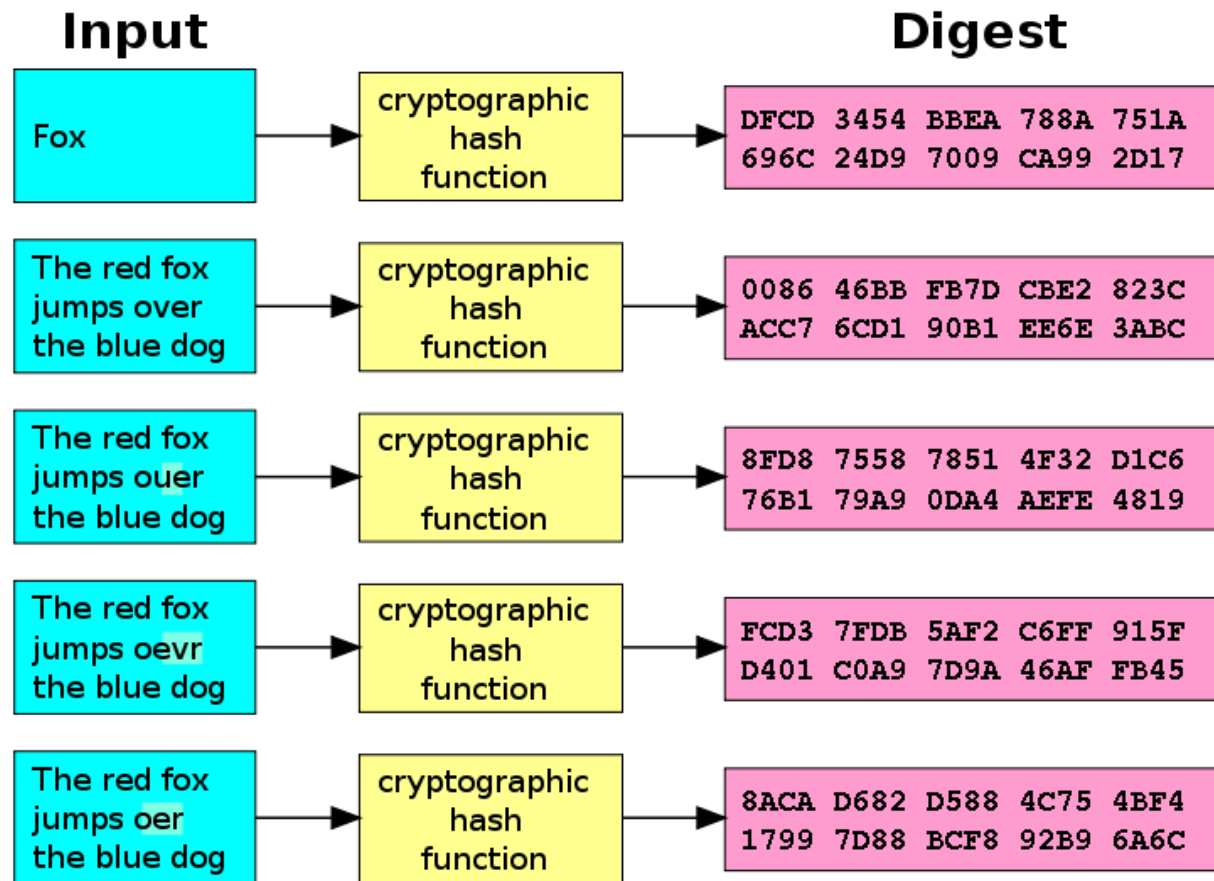
General Trackers Peers HTTP Sources Content

Speed

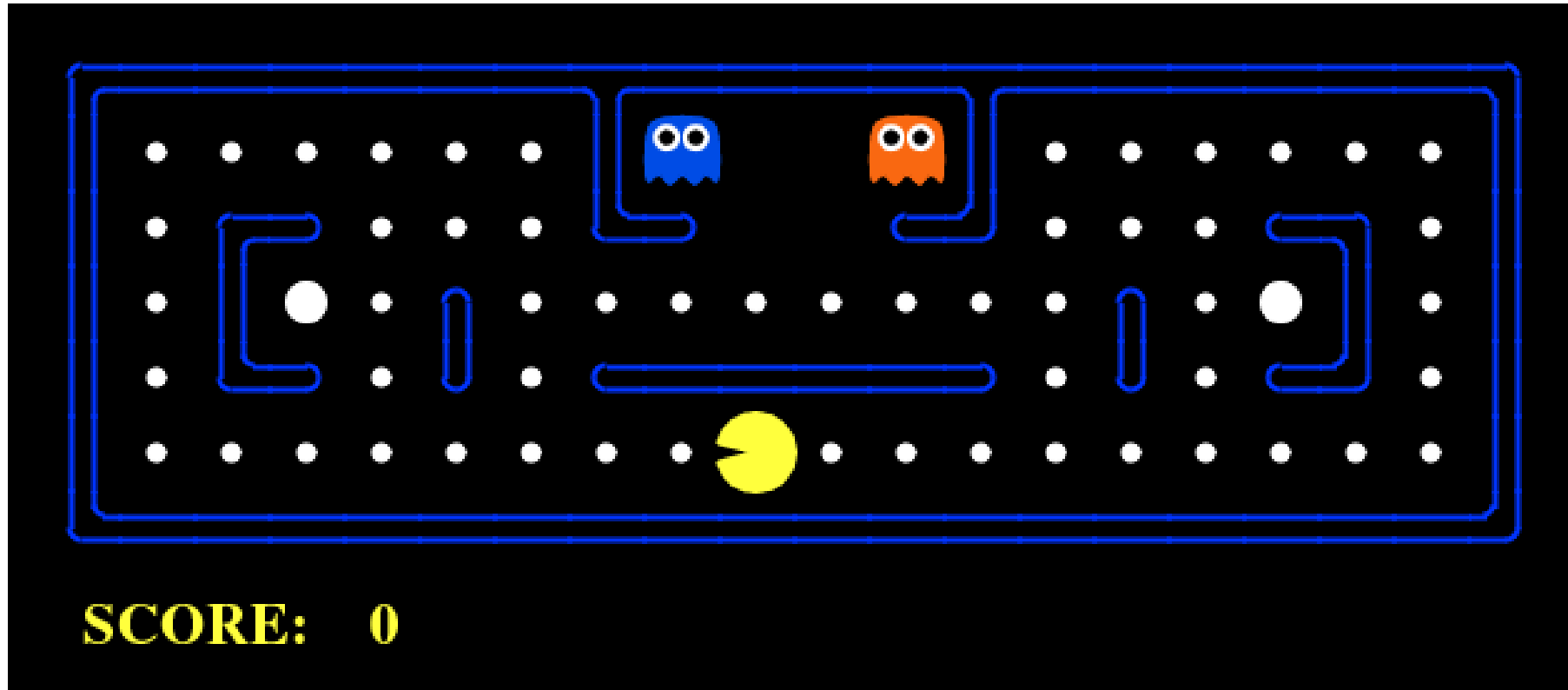
DHT: 360 nodes + 0 B/s (0 B) + 0 B/s (0 B)

2:30 PM 3/26/2018

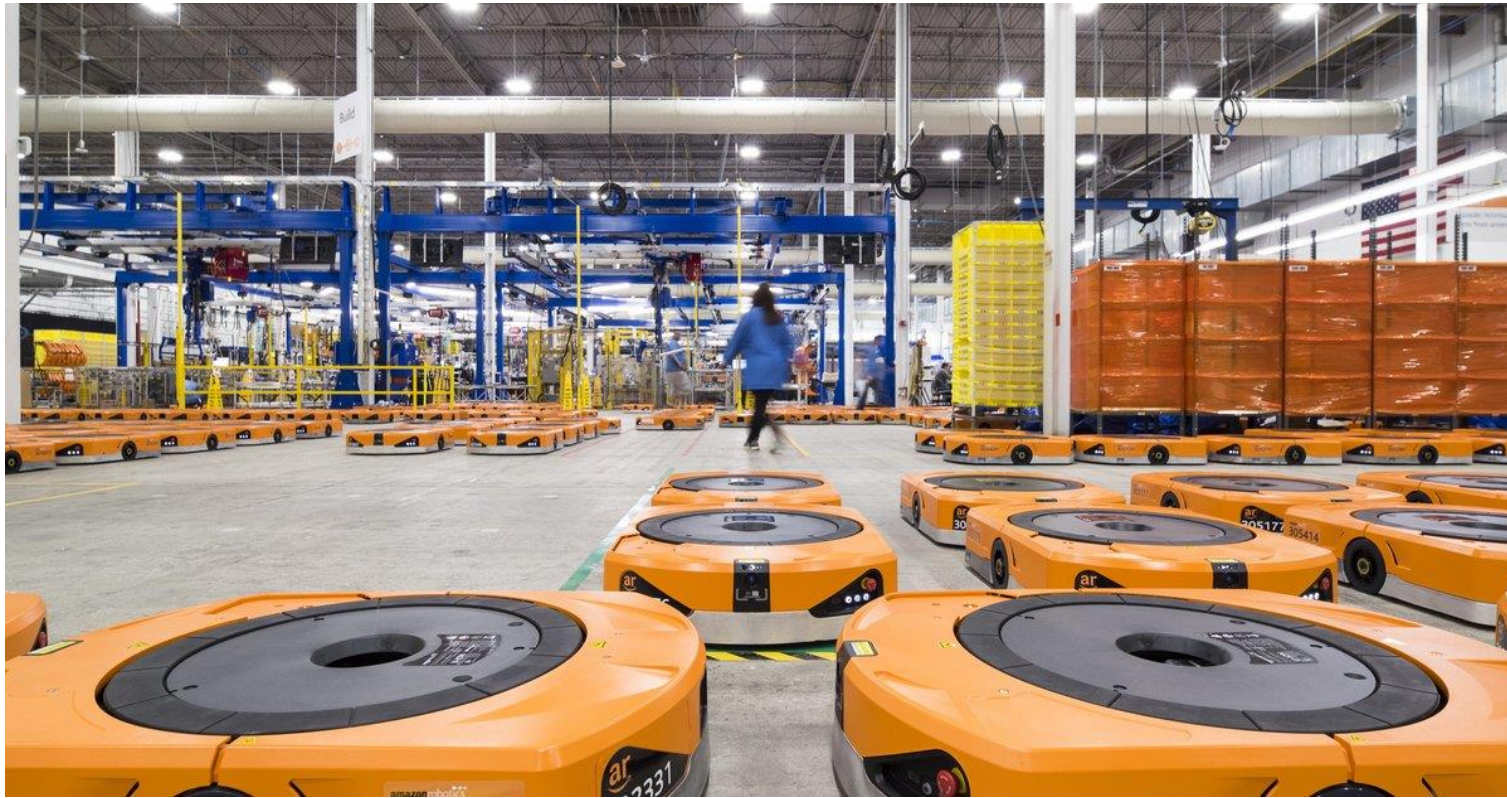
Κρυπτογραφία: cryptographic hash functions



Τεχνητή Νοημοσύνη: Διάφορες Δομές Δεδομένων π.χ., Ουρές
Προτεραιότητας (<http://cgi.di.uoa.gr/~ys02/>,
<https://eclass.uoa.gr/courses/DI517/> <http://ai.di.uoa.gr/>)

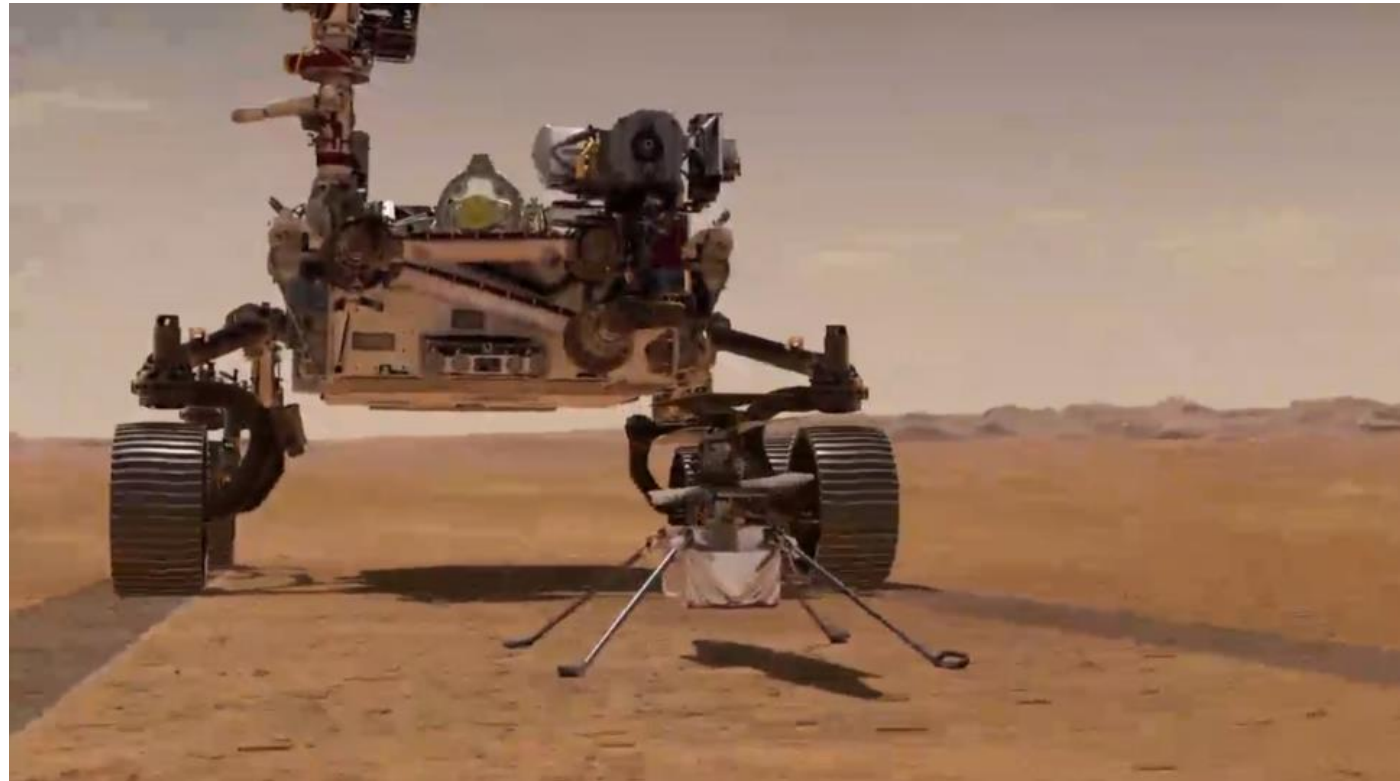


Τεχνητή Νοημοσύνη: Robots in Amazon Warehouses



Perseverance Robot and its Helicopter Exploring Mars

- The helicopter runs linux!



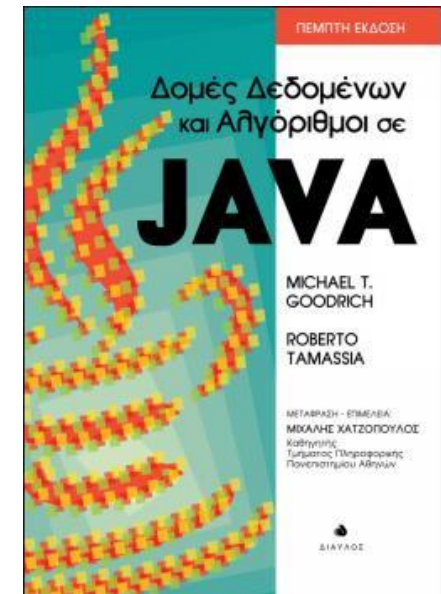
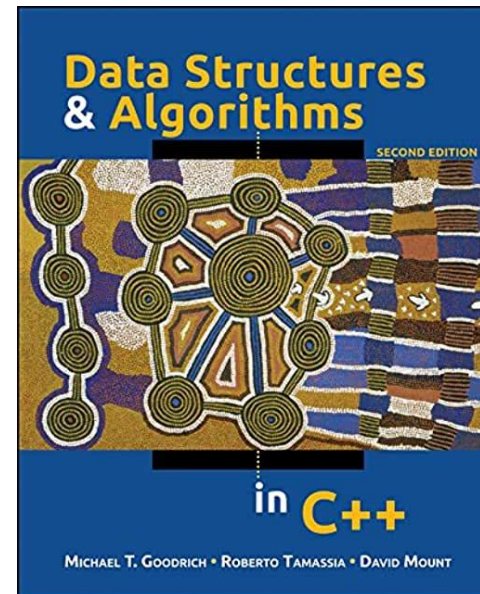
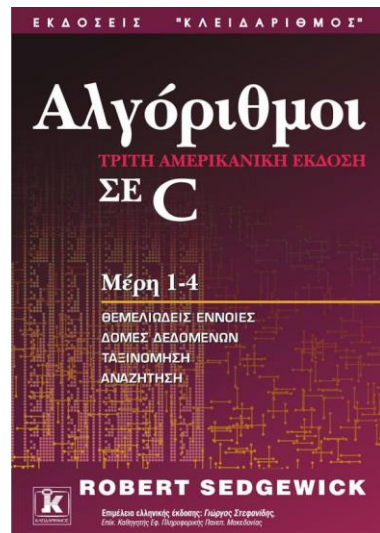
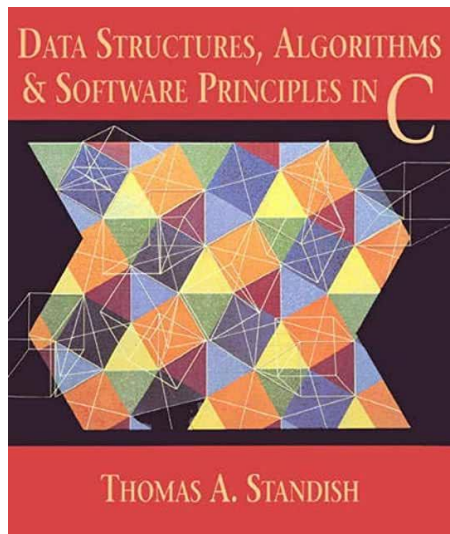
Περιεχόμενο του Μαθήματος

- Συνδεδεμένες αναπαραστάσεις δεδομένων
- Αναδρομή
- Ενότητες και Αφαιρετικοί τυποί Τύποι Δεδομένων στη C.
- Στοίβες
- Ουρές
- Εισαγωγή στην Ανάλυση Αλγορίθμων
- Λίστες και Συμβολοσειρές
- Δένδρα, σωροί, δένδρα εκφράσεων
- Δυαδικά δένδρα αναζήτησης
- Δένδρα AVL
- Δένδρα αναζήτησης m δρόμων, δένδρα 2-3, δένδρα 2-3-4
- Δένδρα κόκκινου-μαύρου
- Δένδρα (a,b), B-δένδρα
- Κατακερματισμός
- Δομές δεδομένων για σύνολα ξένα μεταξύ τους
- Γράφοι
- Γράφοι με βάρη
- Ελάχιστα δένδρα επικάλυψης
- Αλγόριθμοι Ταξινόμησης

Εργαστήριο και Εργασίες

- C, VS Code, linux, gcc, make, github

Βιβλία



Δομές Δεδομένων σε Μοντέρνες Γλώσσες Προγραμματισμού Υψηλού Επίπεδου

- Οι μοντέρνες γλώσσες προγραμματισμού υψηλού επιπέδου (high level programming languages) όπως Python ή Java, μας προσφέρουν έτοιμες πολλές χρήσιμες δομές δεδομένων και **δεν χρειάζεται να τις υλοποιήσουμε όπως θα κάναμε εμείς στη C.**

Java Example with Data Structure Stack

```
import java.util.*;
public class StackDemo {

    static void showpush(Stack st, int a) {
        st.push(new Integer(a));
        System.out.println("push(" + a + ")");
        System.out.println("stack: " + st);
    }

    static void showpop(Stack st) {
        System.out.print("pop -> ");
        Integer a = (Integer) st.pop();
        System.out.println(a);
        System.out.println("stack: " + st);
    }

    public static void main(String args[]) {
        Stack st = new Stack();
        System.out.println("stack: " + st);
        showpush(st, 42);
        showpush(st, 66);
        showpush(st, 99);
        showpop(st);
        showpop(st);
        showpop(st);
        try {
            showpop(st);
        } catch (EmptyStackException e) {
            System.out.println("empty stack");
        }
    }
}
```

Live Demo



Output of the Previous Java Program

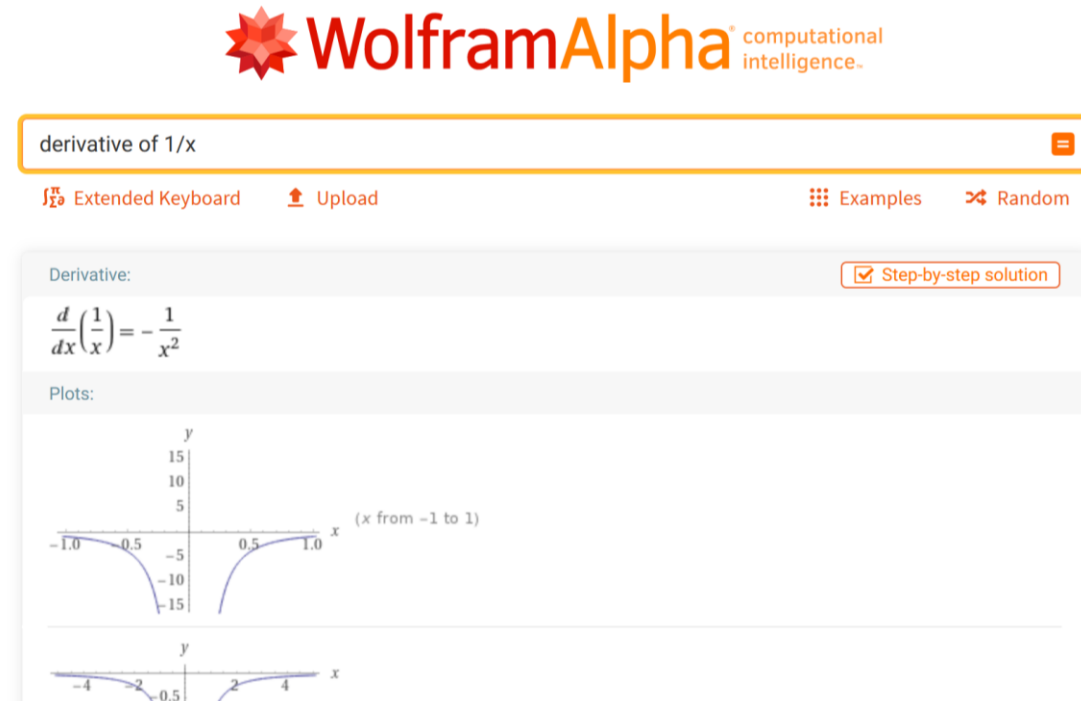
```
stack: [ ]  
push(42)  
stack: [42]  
push(66)  
stack: [42, 66]  
push(99)  
stack: [42, 66, 99]  
pop -> 99  
stack: [42, 66]  
pop -> 66  
stack: [42]  
pop -> 42  
stack: [ ]  
pop -> empty stack
```

Python Example with Data Structure List

```
>>> fruits = ['orange', 'apple', 'pear', 'banana', 'kiwi', 'apple', 'banana']
>>> fruits.count('apple')
2
>>> fruits.count('tangerine')
0
>>> fruits.index('banana')
3
>>> fruits.index('banana', 4) # Find next banana starting a position 4
6
>>> fruits.reverse()
>>> fruits
['banana', 'apple', 'kiwi', 'banana', 'pear', 'apple', 'orange']
>>> fruits.append('grape')
>>> fruits
['banana', 'apple', 'kiwi', 'banana', 'pear', 'apple', 'orange', 'grape']
>>> fruits.sort()
>>> fruits
['apple', 'apple', 'banana', 'banana', 'grape', 'kiwi', 'orange', 'pear']
>>> fruits.pop()
'pear'
```

Γιατί Δομές Δεδομένων με C;

- Για τον ίδιο λόγο που κάνουμε και Μαθηματικά αν και υπάρχει το Wolfram Alpha 😊



Γιατί Δομές Δεδομένων με C;

- Υπάρχουν συστήματα και εφαρμογές που **η αποδοτικότητα τους είναι κρίσιμη** και γι αυτό το λόγο η C και η C++ είναι οι πιο κατάλληλες γλώσσες για την υλοποίηση τους:
 - Operating system kernels (Windows, Linux, Mac OS X, iOS, Android)
 - Database management systems (Oracle, MySQL, MS SQL Server, PostgreSQL)
 - Embedded systems (software systems running in automobiles)
 - ...