



Ανάκτηση Πληροφορίας

Ενότητα 11: Ανάκτηση Πληροφορίας στον Παγκόσμιο Ιστό (Web)

Απόστολος Παπαδόπουλος
Τμήμα Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Περιεχόμενα ενότητας

1. Παγκόσμιος Ιστός.
2. Προκλήσεις.
3. Μηχανές αναζήτησης.
4. Τεχνικές βαθμολόγησης σελίδων
 - i. HITS και PageRank.
5. Τάσεις.



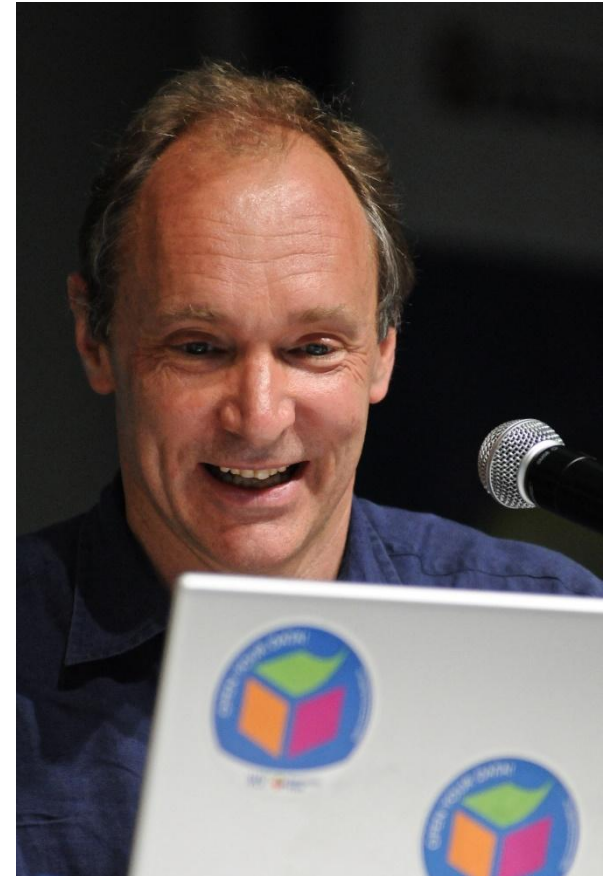


ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Παγκόσμιος Ιστός

Παγκόσμιος Ιστός (The Web)-1

- Ο Παγκόσμιος Ιστός αποτελεί **εφαρμογή** του Internet. Χρησιμοποιεί υπερσυνδέσμους (hyperlinks) για πλοήγηση από σελίδα σε σελίδα (όπως και στο hypertext). HTTP πρωτόκολλο.
- CERN, **Tim Berners-Lee**, 1990. Ανέπτυξε το Web για την ανταλλαγή αρχείων μέσω Internet.



Πηγή: Wikipedia



Παγκόσμιος Ιστός (The Web)-2

- **Ted Nelson**, 1965: δημιουργία του hypertext.
- **Doug Engelbart**, τέλη 60: ανακάλυψη του ποντικιού και η πρώτη υλοποίηση του hypertext.
- **ARPANET**, αρχές 70.
- Αν και η βασική υποδομή υπήρχε στα μέσα της δεκαετίας του 70, χρειάστηκε η είσοδος των **PCs** και **μοντέρνων δικτυακών τεχνικών** για να την έμπνευση του Web.



Παγκόσμιος Ιστός (The Web)-3

- **Mosaic Web Browser, 1993:** δημιουργήθηκε από τους **Marc Andreessen** and **Eric Bina** at **UIUC NCS**.
- Το 1993 δημιουργούνται τα πρώτα **web bots** (spiders).
- Το 1994, οι **David Filo** και **Jerry Yang** φοιτητές του Stanford άρχισαν να συγκεντρώνουν χειροκίνητα τα πιο δημοφιλή web sites σε έναν θεματικό κατάλογο που ονομάστηκε **Yahoo**.



Παγκόσμιος Ιστός (The Web)-4

- Το 1994, ο **Brian Pinkerton** ανέπτυξε τον **WebCrawler** στα πλαίσια μίας εργασίας στο Πανεπιστήμιο της Washington. Στη συνέχεια ο WebCrawler έγινε μέρος του **Excite** και της **AOL**.
- Το 1995, η **DEC** αναπτύσσει τη μηχανή **Altavista**.
- Το 1998, οι **Larry Page** και **Sergey Brin**, διδακτορικοί φοιτητές του Stanford, ξείνησαν την **Google**.





ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Προκλήσεις

Προκλήσεις-1

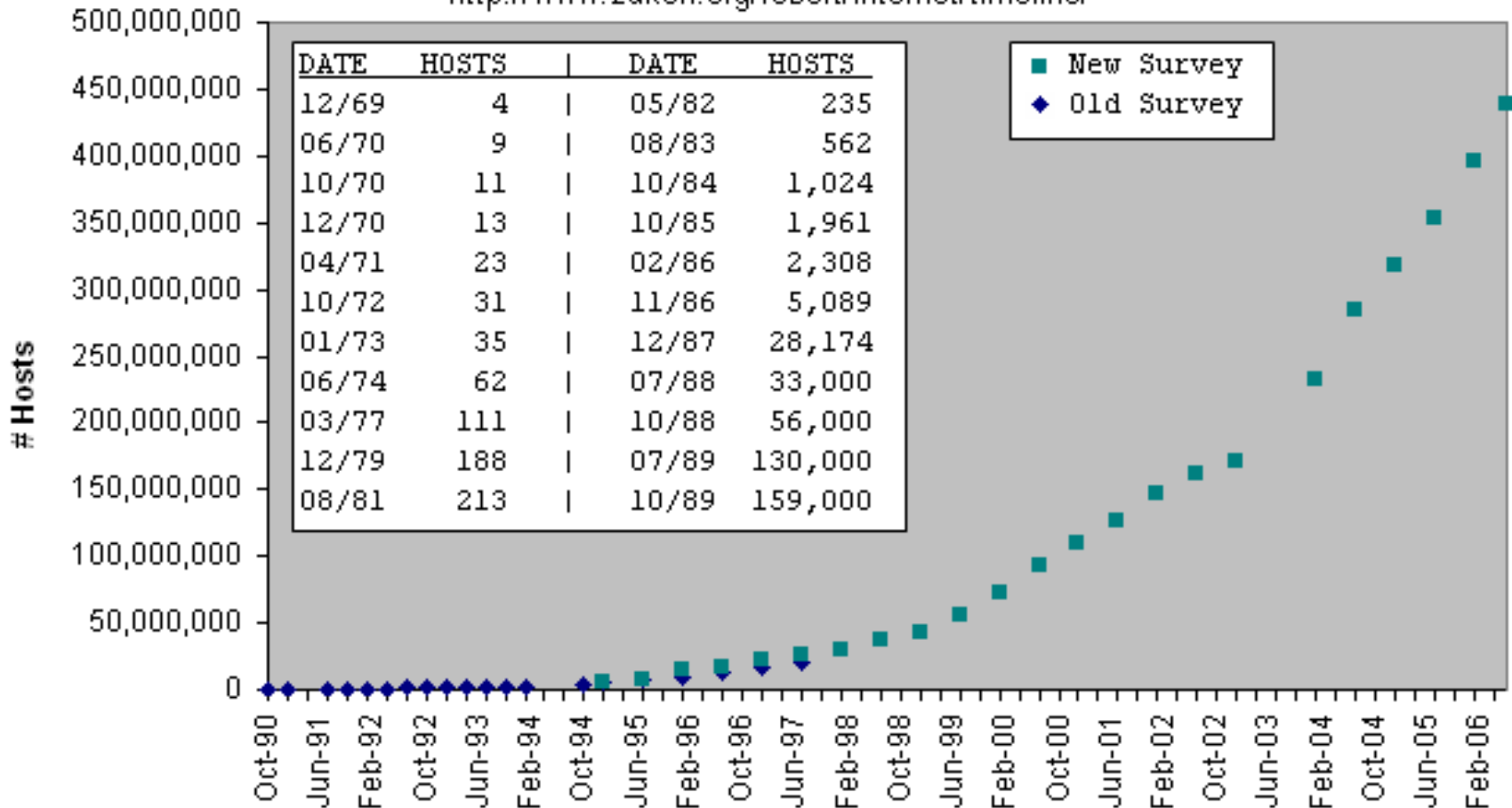
Το Web χαρακτηρίζεται ως η εφαρμογή **killer** για την ανάκτηση πληροφορίας.

- μεγάλες ποσότητες δεδομένων
- κατανομή δεδομένων σε εκατ. sites
- συχνές αλλαγές
- επανάληψη δεδομένων
- πόσο ποιοτικά είναι τα δεδομένα?
- αδόμητα, ημιδομημένα και πλήρως δομημένα δεδομένα
- ετερογένεια (διαφορετικές μορφές, π.χ., html, mpeg, pdf, διαφορετικές γλώσσες, κλπ).



Προκλήσεις-2

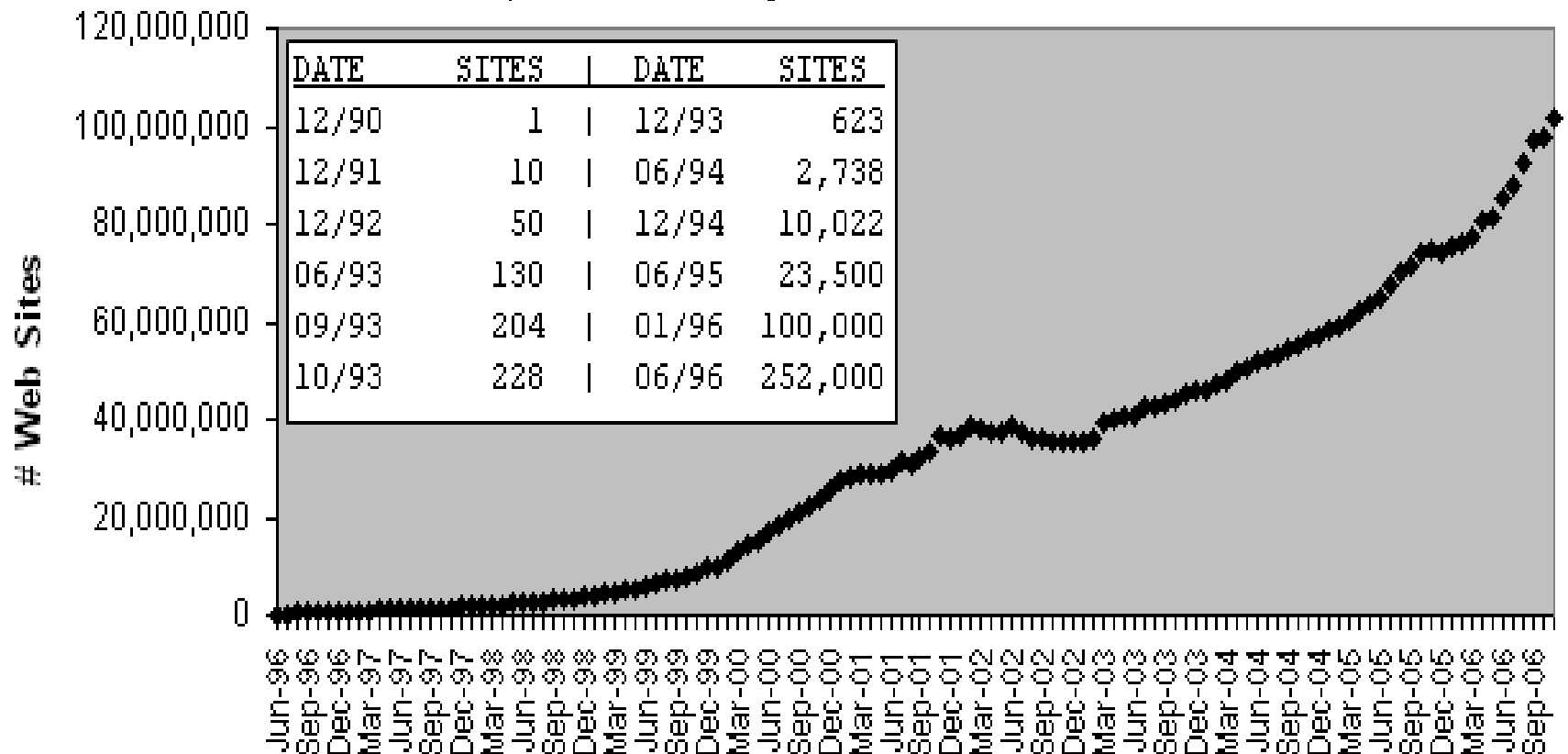
Hobbes' Internet Timeline Copyright ©2006 Robert H Zakon
<http://www.zakon.org/robert/internet/timeline/>



Προκλήσεις-3

Hobbes' Internet Timeline Copyright ©2006 Robert H Zakon

<http://www.zakon.org/robert/internet/timeline/>



Zipf και Web-1

- Ο αριθμός των εισερχομένων και εξερχομένων συνδέσμων μία σελίδα ακολουθεί την **κατανομή Zipf**.
- Το μέγεθος των σελίδων ακολουθεί την **κατανομή Zipf**.
- Ο αριθμός των hits μίας σελίδα ακολουθεί την **κατανομή Zipf**.



Zipf και Web-2

Και ποια είναι η κατανομή Zipf ?

Πολλά δεδομένα του φυσικού κόσμου μπορούν να μοντελοποιηθούν σύμφωνα με την Zipf κατανομή, που είναι ειδική κατηγορία **Power Law**.

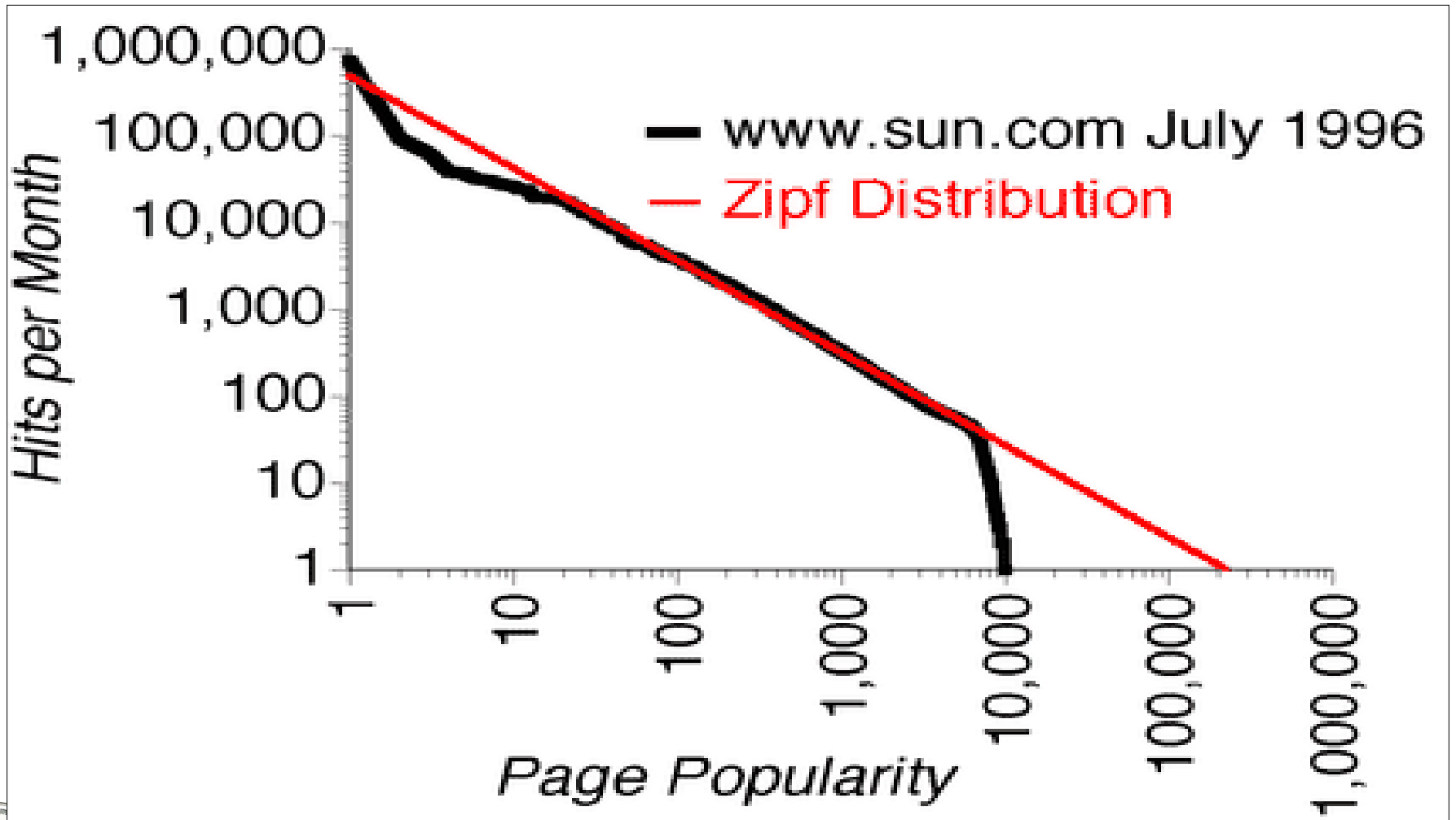
Έστω $F(x)$ ο αριθμός εμφανίσεων της x -οστής συχνότερης λέξης. Ο Zipf Law μας λέει ότι:

$$F(x) \propto \frac{1}{x^a}$$

Το a είναι κοντά στη μονάδα.



Zipf και Web-3



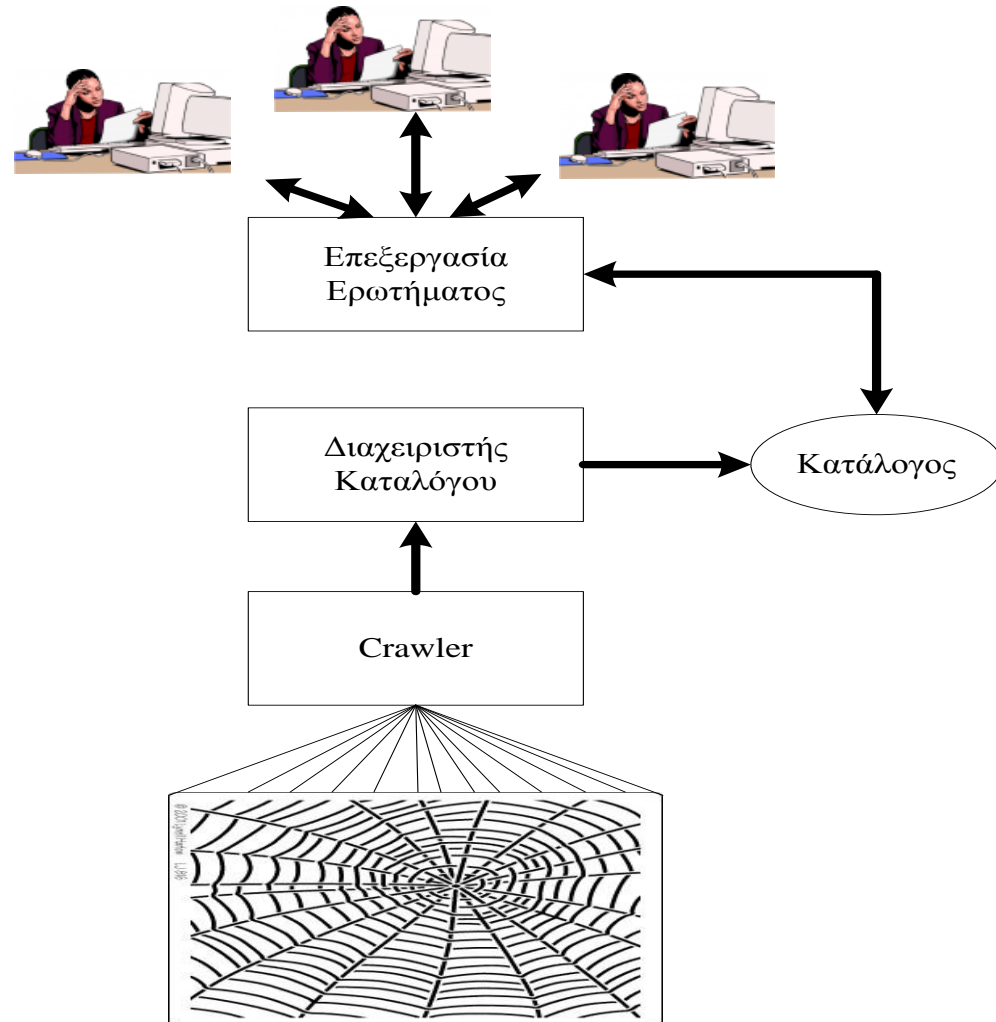


ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Μηχανές Αναζήτησης

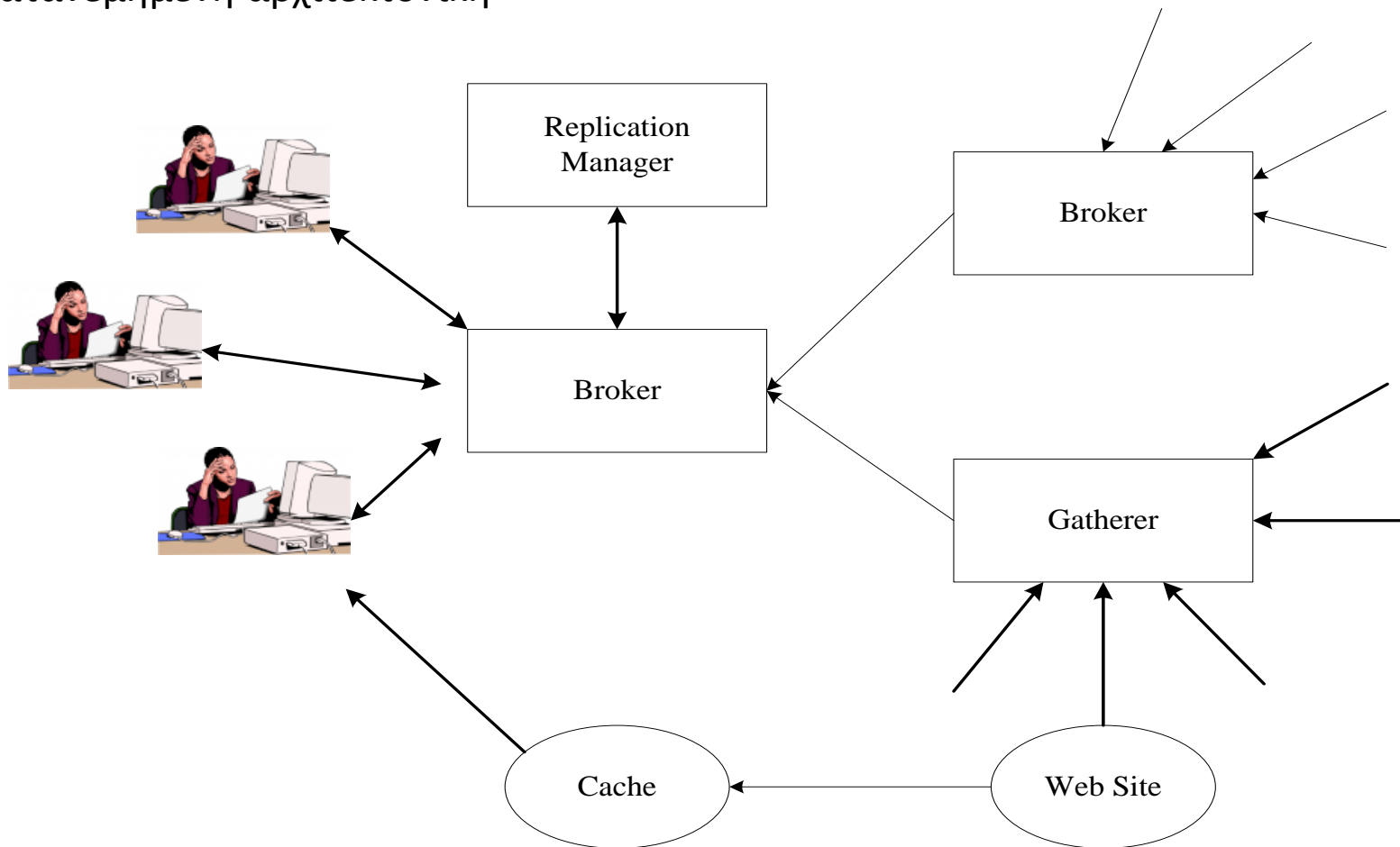
Μηχανές Αναζήτησης-1

- Βασικά στοιχεία.
- Μηχανής αναζήτησης.
- Επεξεργαστής ερωτημάτων.
- Κατάλογος.
- Διαχειριστής καταλόγου.
- Crawler.



Μηχανές Αναζήτησης-2

Κατανεμημένη αρχιτεκτονική



Μηχανές Αναζήτησης: crawling-1

- Web Crawler: ένα πρόγραμμα που διαβάσει («κατεβάζει») σελίδες Web.
- Επίσης καλείται και Spider, Bot, WebBot.
- Προσοχή: ο crawler δεν είναι agent, τρέχει σε κάποιον server και απλά παράγει HTTP αιτήσεις για να «κατεβάσει» τις σελίδες.
- Το ίδιο ακριβώς κάνει και ένας χρήστης χρησιμοποιώντας έναν Web Browser.
- Η διαφορά είναι ότι ο crawler είναι πιο συστηματικός και φυσικά πιο γρήγορος στο browsing.



Μηχανές Αναζήτησης: crawling-2

Απλός αλγόριθμος για crawling

Έστω W το σύνολο των σελίδων που πρέπει να κάνουμε crawling.

1. Αρχικά, επιλέγονται μερικές σελίδες ως αφετηρία και μπαίνουν στο σύνολο W .
2. Διαβάζουμε μία σελίδα p από το σύνολο W . Μετά το parsing της σελίδας p έστω ότι προσδιορίζονται $N(p)$ σελίδες άμεσα επισκέψιμες από την p .
3. Θέτουμε $W := W + N(p) - p$
4. Η διαδικασία επαναλαμβάνεται από το βήμα 1.



Μηχανές Αναζήτησης: crawling-3

Ο αλγόριθμος είναι απλός αλλά ...

- Πως θα κάνουμε crawling τα εκατ. των Web σελίδων;
- Πως θα αποφύγουμε το overloading ενός Web server;
- Τι θα κάνουμε εάν ο ιδιοκτήτης ενός server δεν επιθυμεί crawlers;
- Πως θα χειριστούμε μη έγκυρους συνδέσμους (broken links);
- Με ποιά στρατηγική θα προχωρήσουμε στο crawling (DFS, BFS);
- Τι δομές απαιτούνται για τον έλεγχο των σελίδων που βρίσκουμε;



Μηχανές Αναζήτησης: crawling-4

Πως θα αποφύγουμε τους crawlers;

- Ο διαχειριστής του Web server μπορεί να ορίσει ποιες περιοχές του server δεν πρέπει να διαβαστούν από crawlers, δημιουργώντας ένα κατάλληλα διαμορφωμένο αρχείο με όνομα **robots.txt**
- Επίσης, ο συγγραφέας μίας Web σελίδας μπορεί να δηλώσει ότι δεν επιθυμεί η σελίδα να διαβαστεί και να γίνει indexed από crawlers χρησιμοποιώντας μία ειδική HTML META tag.



Μηχανές Αναζήτησης: crawling-5

<http://gr.yahoo.com/robots.txt>

User-agent: *

Disallow: /gnn

Disallow: /msn

Disallow: /pacbell

Disallow: /pb

Disallow: /bin/query_uk

Rover is a bad dog <<http://www.roverbot.com>>

User-agent: Roverbot

Disallow: /



Μηχανές Αναζήτησης: crawling-6

Χρησιμοποιώντας την META tag **robots** ο δημιουργός μίας σελίδας μπορεί να αποτρέψει την καταλογοποίηση και ανάλυση της σελίδας για links.

Παράδειγμα:

```
<meta name="robots" content="noindex, nofollow">
```

Ο crawler δεν θα πρέπει να κάνει indexing τη σελίδα ούτε και να την αναλύσει για να βρει νέα links.



Μηχανές Αναζήτησης: crawling-7

Για να μπορέσουμε να κάνουμε crawling 1 δις σελίδες το μήνα θα πρέπει να κατεβάζουμε περίπου 400 σελίδες το δευτερόλεπτο.

Άρα, οι δομές δεδομένων θα πρέπει να είναι αποδοτικές.

Επίσης, δεν πρέπει να φορτώνουμε πολύ το server.

Η υλοποίηση του crawler τελικά δεν είναι καθόλου απλή.



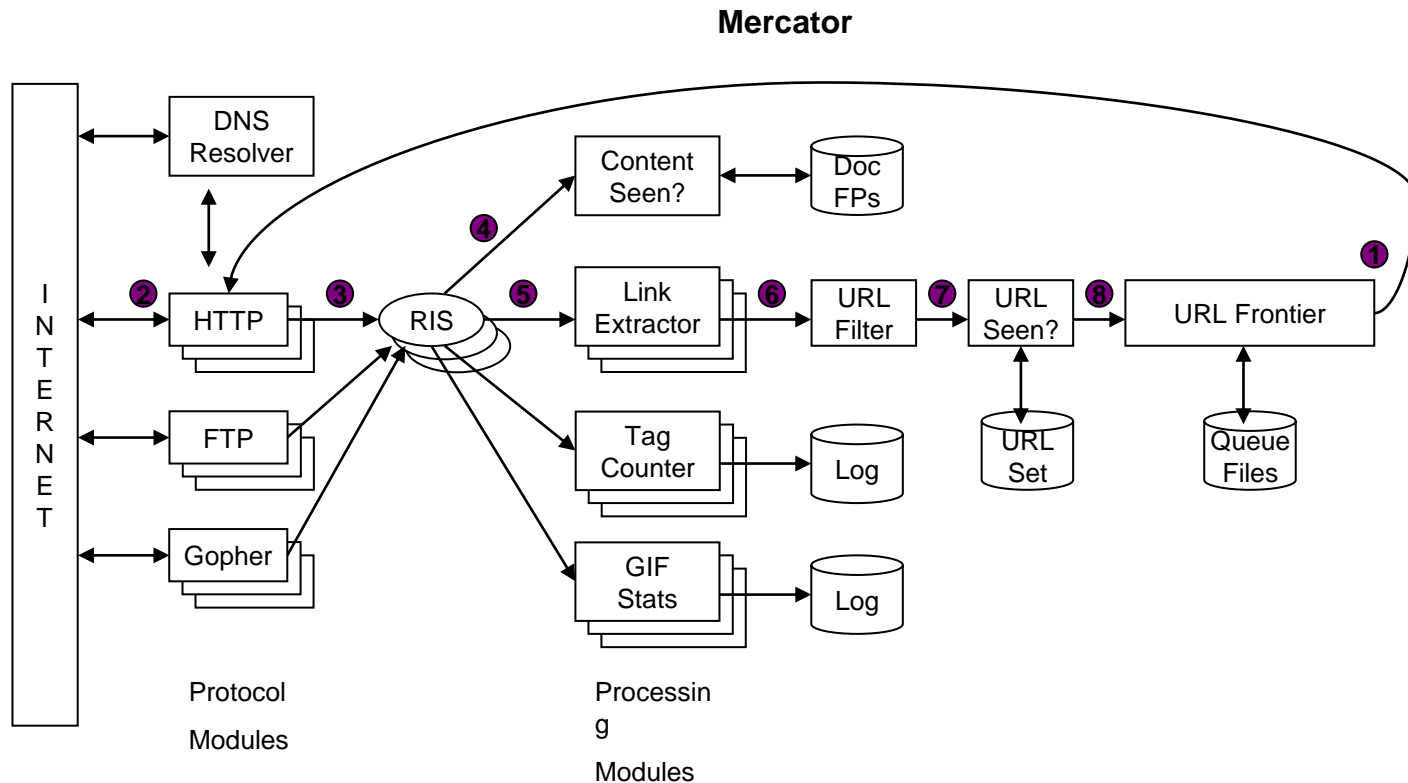
Μηχανές Αναζήτησης: crawling-8

- **Mercator** είναι ένας crawler υψηλών επιδόσεων που δημιουργήθηκε από τους Allan Heydon, Marc Njork, Raymie Stata και συνεργάτες στο Compaq Systems Research Center (αποτελεί συνέχιση της δουλειάς της AltaVista).
- **Heritrix** είναι ένας ανοικτού κώδικα crawler υψηλών επιδόσεων που υλοποιήθηκε από τον Raymie Stata και συνεργάτες στο Internet Archive.



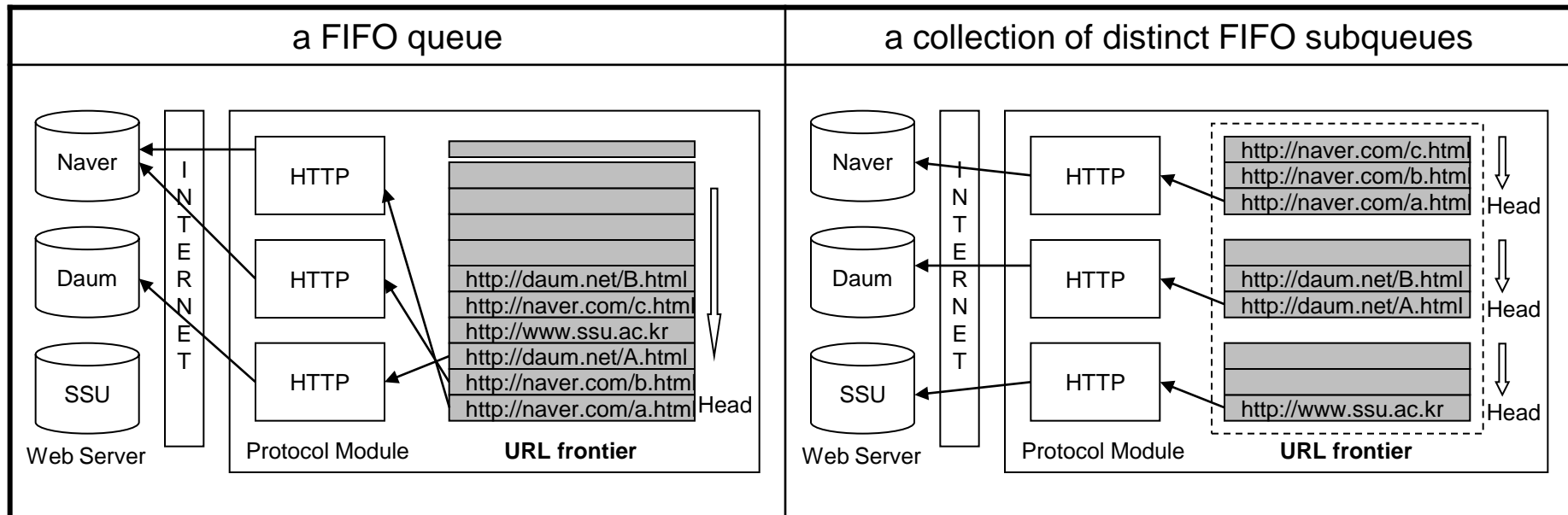
Μηχανές Αναζήτησης: crawling: Mercator

Τα βασικότερα τμήματα του Mercator (thanks to Young Geun Han)

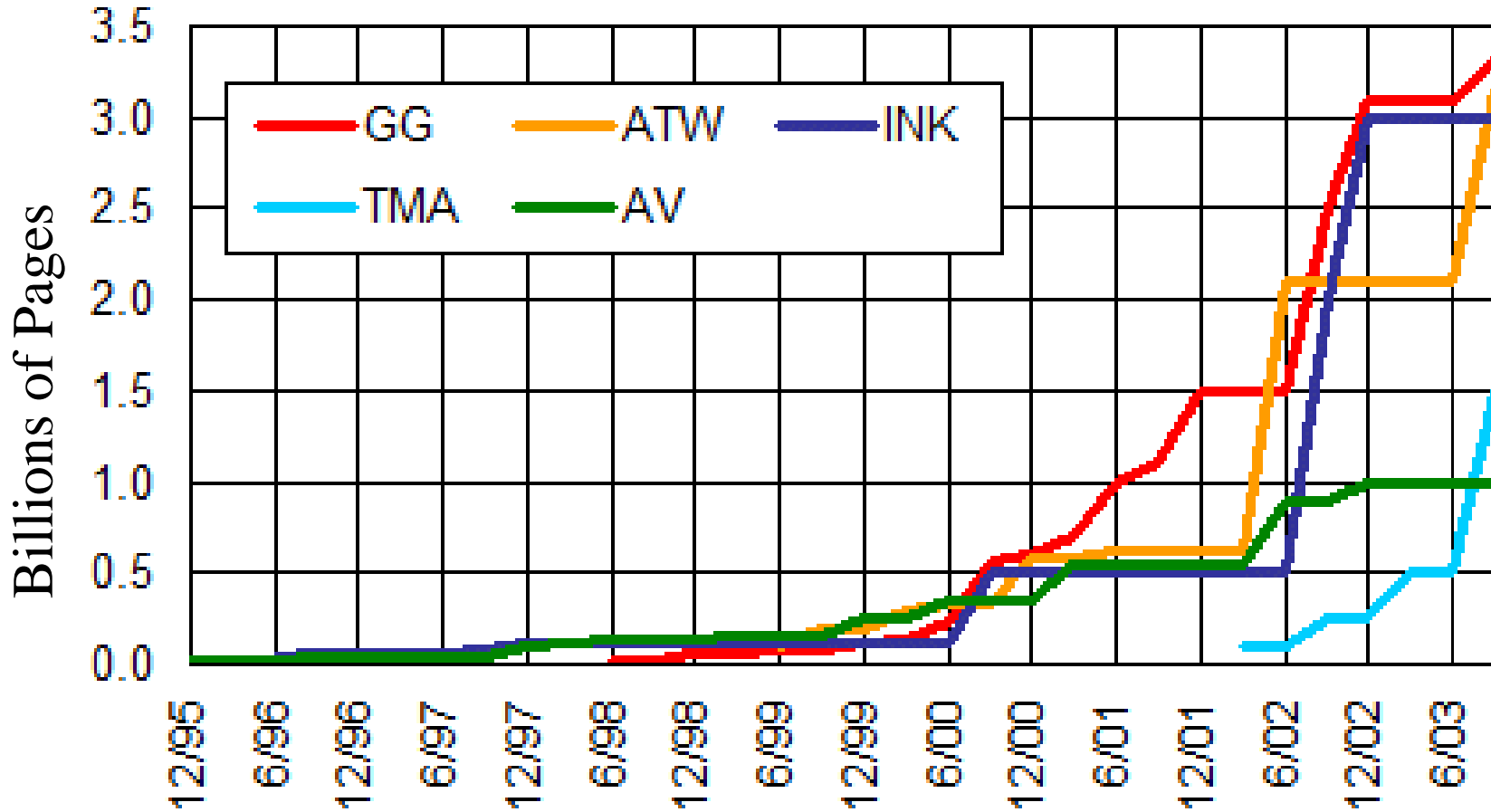


The URL frontier

URL frontier: αποτελεί τη βασική δομή που περιέχει τα URLs που πρέπει να επισκεφτούμε.



Μηχανές Αναζήτησης: indexing-1



Πηγή: <http://searchenginewatch.com/sew/study/2068075/search-engine-sizes>

Μηχανές Αναζήτησης: indexing-2

- Ο κατάλογος μίας μηχανής αναζήτησης είναι συνήθως παραλλαγές του αντεστραμμένου καταλόγου.
- Ωστόσο, η διαχείρισή του απαιτεί τεράστια υπολογιστική ισχύ αν σκεφτούμε ότι οι crawlers διαβάζουν σελίδες με γρήγορους ρυθμούς.





ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Τεχνικές βαθμολόγησης σελίδων

Βαθμολόγηση-1

- Στα κλασικά συστήματα ανάκτησης η βαθμολόγηση των εγγράφων πραγματοποιείται με βάση το περιεχόμενο των σελίδων.
- Για παράδειγμα, αν χρησιμοποιήσουμε τον τύπο του συνημιτόνου έχουμε έναν τρόπο βαθμολόγησης ενός εγγράφου σε σχέση με ένα ερώτημα.
- Γιατί να διαφέρουν τα πράγματα στο Web;



Βαθμολόγηση-2

- Στο Web θα πρέπει να λάβουμε υπόψη μας ότι τα δεδομένα δε χαρακτηρίζονται πάντα από **ποιότητα** και **εγκυρότητα**.
- Η ιδέα είναι να χρησιμοποιήσουμε την πληροφορία των links για να βαθμολογήσουμε μία σελίδα.
- Η τελική βαθμολογία μίας σελίδας προκύπτει από συνδυασμό των επί μέρους βαθμών.



Βαθμολόγηση-3

Ποιά είναι η καλύτερη σελίδα που θα μπορούσε να επισκεφτεί κάποιος που ενδιαφέρεται για Java;

Απάντηση:

- **java.sun.com**
- **comp.lang.java FAQ**

Πως το βρήκαμε;



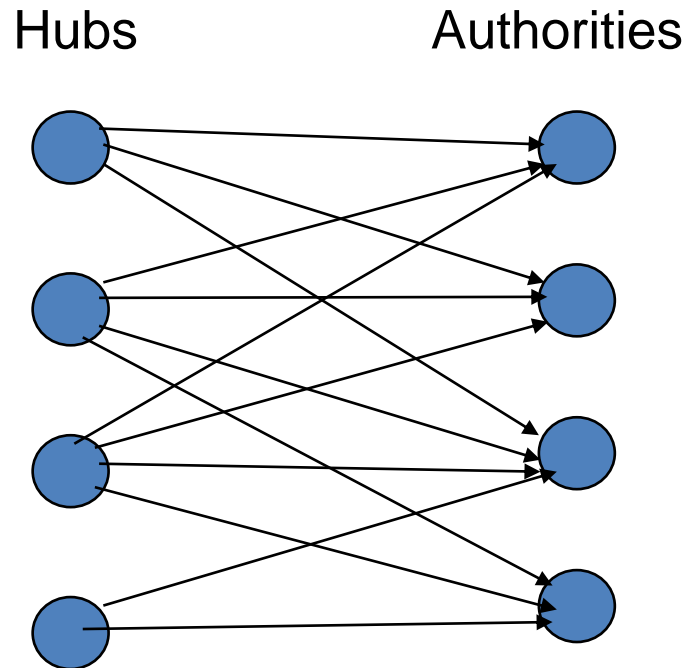
Αλγόριθμος HITS-1

- Ο αλγόριθμος σχεδιάστηκε από τον **Kleinberg** το 1998.
- Προσπαθεί να εντοπίσει **HUBS** και **AUTHORITIES** για ένα δεδομένο θέμα (π.χ., Java).
- **AUTHORITY**: σελίδα που μας δίνει σημαντική, αξιόπιστη και χρήσιμη πληροφορία για ένα θέμα που μας ενδιαφέρει.
- **HUB**: σελίδα που περιέχει links σε σχετικά **AUTHORITIES**.



Αλγόριθμος HITS-2

HUBS και AUTHORITIES σχηματίζουν διμερή γράφο.



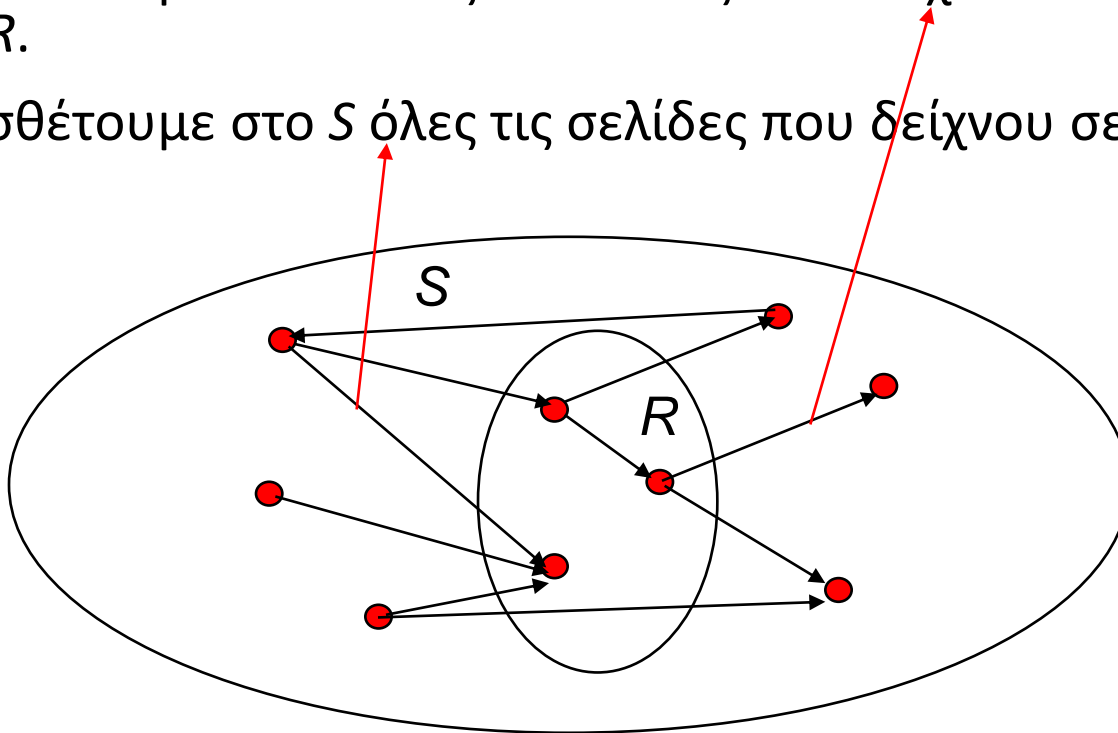
Αλγόριθμος HITS-3

- Ο HITS ξεκινά από ένα σύνολο σχετικών σελίδων, σύνολο R , το οποίο επαυξάνεται για να δημιουργήσει το σύνολο S .
- Στη συνέχεια, ο υπογράφος που ορίζεται από το σύνολο S αναλύεται, ώστε να εντοπιστούν HUBS και AUTHORITIES μέσα στο S .
- Τέλος, τα καλύτερα AUTHORITIES επιστρέφονται στο χρήστη.



Αλγόριθμος HITS-4

- Για ένα ερώτημα Q , έστω R το σύνολο των σελίδων που επιστρέφονται από μία μηχανή αναζήτησης.
- Αρχικοποιούμε $S := R$.
- Προσθέτουμε στο S όλες οι σελίδες που δείχνονται από σελίδες του R .
- Προσθέτουμε στο S όλες τις σελίδες που δείχνου σε σελίδες του R .



Αλγόριθμος HITS-5

- Ακόμη και μέσα στο σύνολο S , οι κόμβοι (σελίδες) με μεγάλο βαθμό εισόδου δεν είναι κατ' ανάγκη authorities. Μπορεί απλά να είναι δημοφιλείς σελίδες (π.χ., Amazon).
- Για να βρούμε τα πραγματικά authorities θα πρέπει να δούμε από πόσα hubs δείχνεται η κάθε σελίδα.
- Ο αλγόριθμος εντοπίζει για κάθε σελίδα το πόσο hub και πόσο authority είναι.



Αλγόριθμος HITS-6

- Επαναληπτικός αλγόριθμος που εκτελείται μέχρι να συγκλίνει.
- Για κάθε σελίδα $p \in S$ κρατάμε τα εξής:
 - authority score: $as(p)$ (vector \mathbf{a})
 - hub score: $hs(p)$ (vector \mathbf{h})
- Αρχικά για κάθε $p \in S$, $as(p) = hs(p) = 1$
- Τα scores κανονικοποιούνται έτσι ώστε:

$$\sum_{p \in S} as(p)^2 = 1 \qquad \sum_{p \in S} hs(p)^2 = 1$$



Αλγόριθμος HITS-7

- Τα authorities δείχνονται από πολλά καλά hubs:

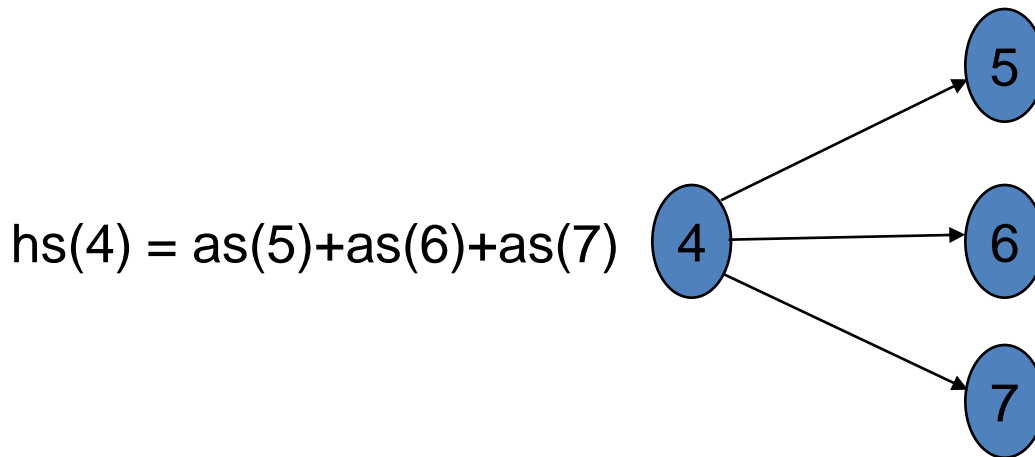
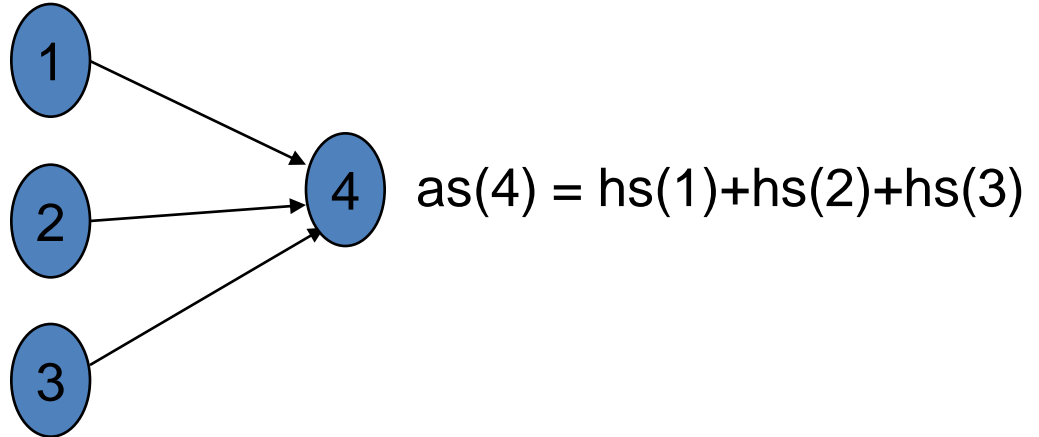
$$as(p) = \sum_{q:q \rightarrow p} hs(q)$$

- Τα hubs δείχνουν σε πολλά καλά authorities:

$$hs(p) = \sum_{q:p \rightarrow q} as(q)$$



Αλγόριθμος HITS-8



Αλγόριθμος HITS-9

Αρχικοποίηση για κάθε $p \in S$: $as(p) = hs(p) = 1$

for $i = 1$ to k :

for all $p \in S$: $as(p) = \sum_{q:q \rightarrow p} hs(q)$ (ενημέρωση authority scores)

for all $p \in S$: $hs(p) = \sum_{q:p \rightarrow q} as(q)$ (ενημέρωση hub scores)

for all $p \in S$: $as(p) = as(p)/c$ $c: \sum_{p \in S} as(p) = 1$

for all $p \in S$: $hs(p) = hs(p)/c$ $c: \sum_{p \in S} hs(p) = 1$



Αλγόριθμος HITS-10

- Ο αλγόριθμος συγκλίνει σε σταθερό σημείο αν επαναληφθεί πάρα πολλές φορές.
- Έστω A ο πίνακας γειτνίασης του υπογράφου που ορίζεται από το σύνολο S .
 - $A_{ij} = 1$ για $i \in S, j \in S$ iff $i \rightarrow j$ (η i δείχνει στην j)
- Το διάνυσμα as συγκλίνει στο σημαντικότερο ιδιοδιάνυσμα του πίνακα $A^T A$
- Το διάνυσμα hs συγκλίνει στο σημαντικότερο ιδιοδιάνυσμα του πίνακα $A A^T$
- Στην πράξη, με 20 επαναλήψεις έχουμε ικανοποιητικά αποτελέσματα.



Αλγόριθμος HITS-11

- Authorities for query: “Java”
 - java.sun.com
 - comp.lang.java FAQ
- Authorities for query “search engine”
 - Yahoo.com
 - Excite.com
 - Lycos.com
 - Altavista.com
- Authorities for query “Gates”
 - Microsoft.com
 - roadahead.com



Αλγόριθμος PageRank-1

- Εναλλακτικός τρόπος ανάλυσης συνδέσμων, **Brin** και **Page**, 1998 (Google).
- Δεν κάνει διάκριση μεταξύ hubs και authorities.
- Η βαθμολογία των σελίδων στηρίζεται μόνο σε authorities.
- Εφαρμόζεται σε όλο το Web και όχι μόνο σε μία γειτονιά όπως ο HITS.
- Ο αλγόριθμος είναι **query independent** σε αντίθεση με τον HITS που είναι query dependent.



Αλγόριθμος PageRank-2

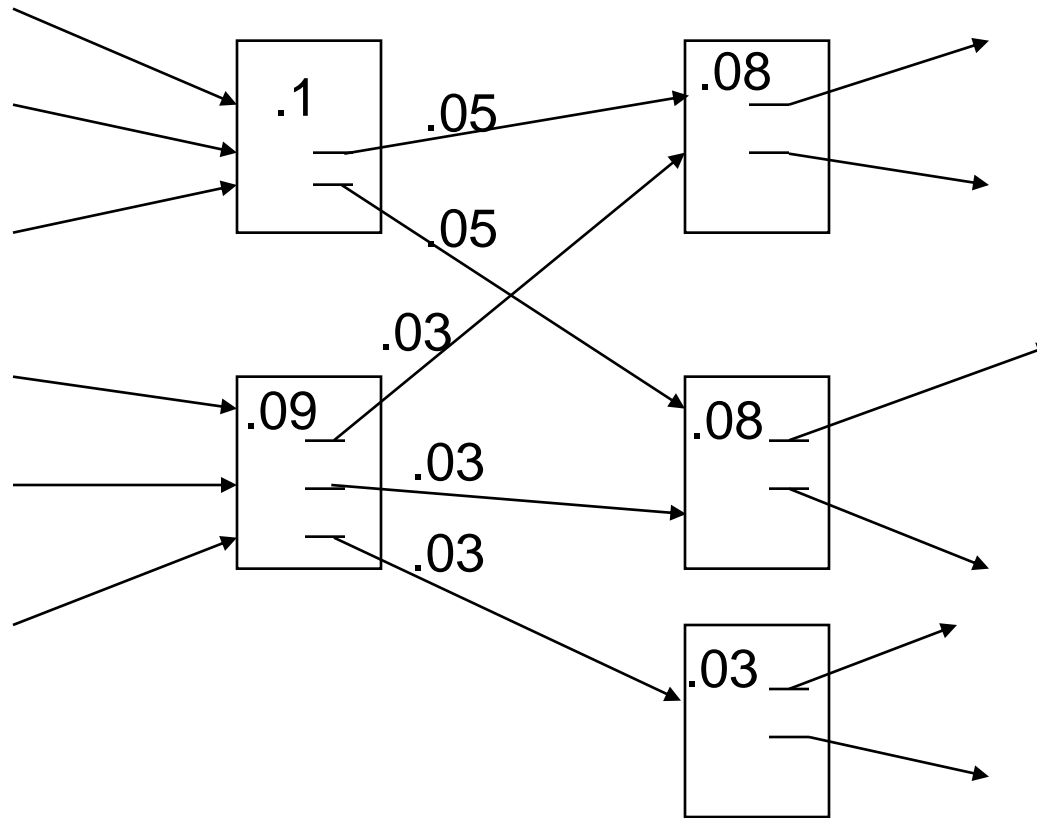
- Χρησιμοποιώντας μόνο το πλήθος των εισερχομένων ακμών δεν μπορούμε να βγάλουμε ασφαλές συμπέρασμα.
- Εξίσωση για κάθε σελίδα p του Web:

$$R(p) = c \sum_{q:q \rightarrow p} \frac{R(q)}{N_q}$$

- N_q αριθμός εξερχομένων ακμών από τη σελίδα q .
- c χρησιμοποιείται για κανονικοποίηση ώστε το άθροισμα των βαθμών (ranks) των σελίδων να είναι 1.



Αλγόριθμος PageRank-3



Αλγόριθμος PageRank-4

Έστω S το σύνολο των σελίδων

Αρχικοποίηση $\forall p \in S: R(p) = 1/|S|$

repeat έως ότου οι βαθμοί δεν αλλάζουν πολύ

$\forall p \in S:$

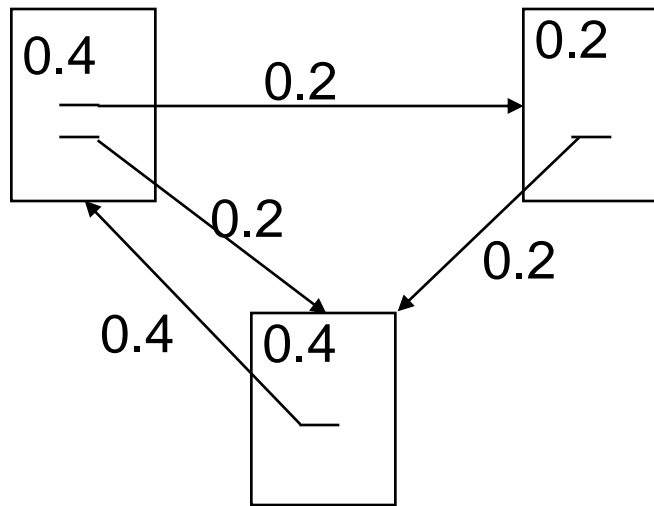
$$R'(p) = \sum_{q:q \rightarrow p} \frac{R(q)}{N_q}$$

$$c = 1 / \sum_{p \in S} R'(p)$$

$$\forall p \in S: R(p) = cR'(p)$$

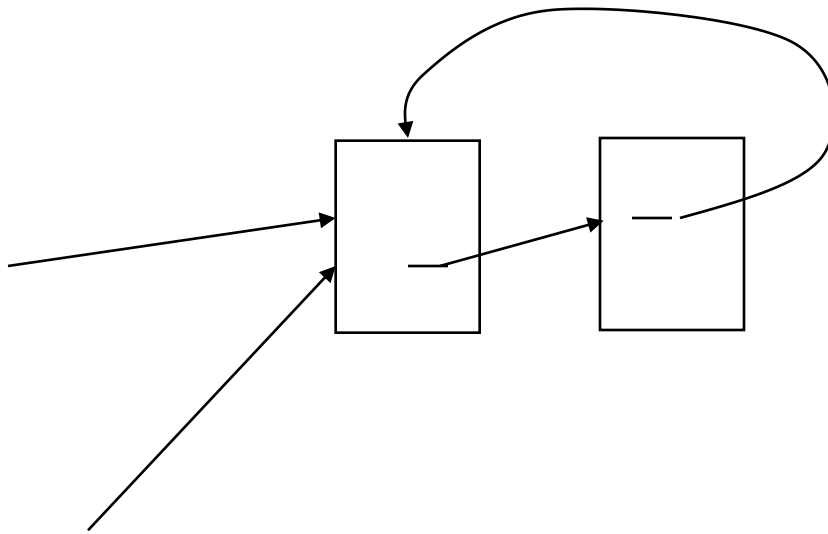


Αλγόριθμος PageRank-5



Αλγόριθμος PageRank: πρόβλημα

Πρόβλημα με τον απλό αλγόριθμο



Ο βαθμός καταναλώνεται

Αλγόριθμος PageRank: λύση

Εισάγουμε μία πηγή βαθμού E που συνεχώς δίνει ένα σταθερό κομμάτι βαθμού $E(p)$.

$$R(p) = c \left(\sum_{q:q \rightarrow p} \frac{R(q)}{N_q} + E(p) \right)$$



Αλγόριθμος PageRank-6

- Έστω S ο συνολικός αριθμός σελίδων.
- $\forall p \in S: E(p) = \alpha / |S|$ (για $0 < \alpha < 1$, π.χ., **0.15**)
- Αρχικοποίηση $\forall p \in S: R(p) = 1 / |S|$
- Μέχρι σύγκλιση

$$\forall p \in S:$$

$$R'(p) = \left[(1 - \alpha) \sum_{q:q \rightarrow p} \frac{R(q)}{N_q} \right] + E(p)$$

$$c = 1 / \sum_{p \in S} R'(p)$$

$$\forall p \in S: R(p) = cR'(p)$$



Αλγόριθμος PageRank-7

- Τα πρώτα πειράματα έγιναν με περίπου 322 εκατ links.
- Ο PageRank έδειξε ότι συγκλίνει σε περίπου 52 επαναλήψεις.
- Εμπειρικά, απαιτούνται $O(\log n)$ επαναλήψεις για σύγκλιση, όπου n το πλήθος των links.



Σύγκριση

PageRank

- Ακριβός στους υπολογισμούς
- Εφαρμόζεται μία φορά για όλες τις σελίδες
- Ανεξάρτητος ερωτήματος
- Ο βαθμός μίας σελίδας υπολογίζεται ως προς το authority

HITS

- Ακριβός στους υπολογισμούς
- Εφαρμόζεται για κάθε ερώτημα
- (άρα εξαρτάται από το ερώτημα)
- Δίνει hubs και authorities
- Η ποιότητα του αποτελέσματος εξαρτάται από την ποιότητα του αρχικού συνόλου σελίδων



Η Βαθμολόγηση της Google

- Με βάση παλαιότερες δημοσιεύσεις:
 - Ομοιότητα βάσει διανυσματικού μοντέλου.
 - Ομοιότητα βάσει γειτονικότητας των λέξεων.
 - Ομοιότητα με βάρη στα HTML-tags.
 - Βαθμός από PageRank.
- Δεν έχουμε διαθέσιμες λεπτομέρειες για το πώς ακριβώς γίνεται σήμερα η βαθμολόγηση από την Google (για προφανείς λόγους).





ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Τάσεις

Τάσεις

- Κατανεμημένες μηχανές αναζήτησης.
- P2P μηχανές αναζήτησης.
- Multimedia μηχανές αναζήτησης (audio, images, video).
- Εξόρυξη δεδομένων από δεδομένα Web (web mining).
- Continuous ερωτήματα.



Χρήσιμοι Σύνδεσμοι

Hobbe's Internet Timeline

<http://www.zakon.org/robert/internet/timeline>



Χρήσιμα Άρθρα

- S. Brin, L. Page, “The Anatomy of a Large Scale Hypertextual Web Search Engine”, *Computer Networks and ISDN Systems*, 1998.
- M. Faloutsos, P. Faloutsos, and C. Faloutsos. “On Power Law Relationships of the Internet Topology”, *ACM SIGCOMM*, 1999.
- R. Kumar, P. Raghavan, S. Rajagopalan D. Sivakumar, et al, “The Web as a Graph”, *ACM PODS*, 2000.



Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Απόστολος Παπαδόπουλος. «Ανάκτηση πληροφορίας. Ανάκτηση Πληροφορίας στον Παγκόσμιο Ιστό (Web Information Retrieval)». Έκδοση: 1.0. Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση:
<http://eclass.auth.gr/courses/OCRS388/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Μη Εμπορική Χρήση - Όχι Παράγωγα Έργα 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>





Τέλος ενότητας

Επεξεργασία: <Μαυρίδης Απόστολος>
Θεσσαλονίκη, <Εαρινό εξάμηνο 2013-2014>





ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Σημειώματα

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

