# Linguaggi e Traduttori
## a.a. 2005/2006
## Tesina n° 7

## **Titolo:** Motore di ricerca
## Descrizione
Si intende realizzare un programma che simula un motore di ricerca. Il programma deve parsificare una serie si pagine html con lo scopo di ricavare alcuni parametri importanti per l'indicizzazione.
In primo luogo dovranno essere identificati i principali tag html: *<h1>, <h2>, <h3>, <h4>, <h5>, , <i>, <u>, <p>, <pre>, <a>, <img>, <meta>, <title>, <ul>, <li>, <ol>, <table>, <tr>, <td>, <html>, <head>, <body>,…*

Il parser, partendo da un file html iniziale passato da linea di comando, dovrà individuare tutti i link presenti nel file (indicati dal tag *<a>*) e andare ad aprire ricorsivamente tutte le pagine puntate da tali link.
**Es.**
*<a href="http://www.skenz.it/esercitazioni/Esercitazione1.html" name="eser1">testo</A>*
Dovrà essere aperta la pagina: *esercitazioni/Esercitazione1.html*
Si fa notare che sono molto probabili le presenze di cicli, dovrà quindi essere mantenuta una struttura in memoria che tiene traccia delle url già analizzate.

Per ogni pagina aperta dovrà essere compiuta un'indicizzazione.
Per ognuno di questi tag:
*<h1>, <h2>, <h3>, <h4>, <h5>, , <i>, <u>, <p>, <pre>, <a>, <img>, <meta>, <title>*
verrà dato un punteggio:

| | |
|---|---|
| *<h1>* | *10* |
| *<h2>* | *9* |
| *<h3>* | *8* |
| *<h4>* | *7* |
| *<h5>* | *6* |
| *<a>* | *10* |
| ** | *+3* |
| *<i>* | *+2* |
| *<u>* | *+1* |
| *<p>* | *3* |
| *<pre>* | *3* |
| *<a>* | *15* |
| *<title>* | *20* |

Il punteggio del testo normale non contenuto in nessuno dei tag è di *2*.
Ogni volta che vengono incontrate delle parole all'interno di un tag dovranno essere memorizzate in una struttura con associato il punteggio dovuto al tag che le contiene.
Fanno eccezione i tag *, <i>, <u>* i quali, essendo utilizzati all'interno di altri tag, sono dei modificatori di punteggio: il tag **, ad esempio, fa aumentare, per le parole contenute in tale tag, il punteggio del tag più esterno di 3 punti.
**Es.**
*asterix<h1>obelix idefix</h1>obelix*
A seguito dell'indicizzazione dovranno essere salvati in memoria i seguenti punteggi:
*idefix 13* (10+(+3))
*obelix 12* (10+2)

*asterix 2*

Esiste il tag <*meta*> che ha un comportamento leggermente diverso rispetto agli altri tag. Il tag <*meta*> è stato creato appositamente per facilitare i motori di ricerca nell'indicizzazione fornendo appunto dei metadati. In questa banale implementazione di un motore di ricerca il tag <*meta*> verrà utilizzato alla stregua degli altri tag, utilizzando le parole in esso contenute come delle normali parole appartenenti al file html.
Il tag <*meta*> ha la seguente sintassi:
*<meta name="<nome_del_tag>" content="<informazione_contenuta_nel_tag>">*
<*nome_del_tag*> rappresenta il nome del metadato, le cui informazioni sono contenute nel campo *content.*
Si richiede che vengano identificati i seguenti tipi di metadati con i corrispettivi punteggi associati:
*Title 20*
*Author 1*
*Subject 12*
*Description 8*
*Keywords 18*
**Es.**
Dato un file html che contiene i seguenti tag:
*<META NAME="Title" CONTENT="Linguaggi e traduttori">*
*<META NAME="Keywords" CONTENT="jflex cup grammatiche linguaggi scanner parser">*
dovranno essere forniti i seguenti punteggi:
*linguaggi 38*
*traduttori 20*
*e 20*
*jflex 18*
*cup 18*
*parser 18*
*scanner 18*
*grammatiche 18*

Un altro parametro molto importante utilizzato e brevettato dal motore di ricerca google è il page ranking (PR). Il PR è un parametro che viene fornito per ogni pagina web e indica l'importanza della pagina. Nella sua implementazione più banale, che è quella che si intende realizzare nella tesina, è definito come il numero di link di pagine esterne che puntano alla pagina sulla quale si vuole calcolare il PR.
**Es.**
Date le seguenti pagine e i link presenti all'interno di ogni pagina:
*pagina1 -> a:pagina2 a:pagina3*
*pagina2 -> a:pagina1*
*pagina3 -> a:pagina1 a:pagina2*
*pagina4 -> a:pagina1*
si otterranno i seguenti valori di PR:
*pagina1: 3*
*pagina2: 2*
*pagina3: 1*
*pagina4: 0*
Per ogni pagina analizzata dovrà perciò essere calcolato il PR.

Il programma dovrà terminare fornendo un output simile a quello di seguito riportato:
*index.html (PR: 10)*
*linguaggi 38*
*traduttori 20*
*e 20*
*jflex 18*
*cup 18*

*esercitazioni/Esercitazione1.html (PR: 8)*
*e 20*
*jflex 18*
*cup 18*

*pagina2.html (PR: 2)*
*linguaggi 38*
*traduttori 20*
*e 20*
*jflex 18*
*cup 18*
*parser 18*
*scanner 18*
*grammatiche 18*
L'output dovrà seguire l'ordine di PR decrescente per le pagine e di punteggio decrescente per le parole indicizzate.

Assieme al programma dovrà essere allegata una breve relazione contenente una veloce descrizione delle strutture dati utilizzate, di come il programma funziona e delle specifiche/limiti di funzionamento. Dovranno poi essere allegati dei **file di esempio** per verificare il corretto funzionamento del programma.