

Algoritmica 2

Appello 2: 02/06/2010

Esercizio 1

[punti 7]

Scelto un problema combinatorio a piacere, presentare un algoritmo non deterministico per risolverlo redatto in un linguaggio tipo Java cui siano aggiunte le primitive: **success**, **failure**, **choice(S)** ove S è un insieme arbitrario.

Spiegare con precisione come si valuta la complessità dell'algoritmo nel modello non deterministico.

Esercizio 2

[punti 7]

a) Definire le formule di ranking per il PageRank e HITS, e commentare le loro differenze.

b) Applicare la formula del PageRank per un solo passo a un grafo di 3 pagine web costituito dai seguenti link orientati $\{(1,3), (2,3), (3,2)\}$, con pesi iniziali delle pagine uguali a 1 e probabilità di teletrasporto pari a $q = \frac{1}{2}$.

c) Applicare la formula di HITS per un solo passo al grafo precedente assumendo che questo sia il base set di una generica query, e si assumano score iniziali per $a()$ e $h()$ uguali a 1.

Esercizio 3

[punti 7]

a) Si descrivano molto sinteticamente le caratteristiche comuni agli algoritmi di Knuth-Morris-Pratt e di Boyer-Moore.

b) Si descrivano le differenze tra essi in termini

b1) del preprocessing che fanno e

b2) di come effettuano la ricerca nel testo

c) Si mostrino un semplice testo e un semplice pattern che costituiscano il caso pessimo in termini di numero di shift di uno dei due algoritmi sopra citati, motivando la risposta.

Esercizio 4

[punti 7]

Dimostrare che l'algoritmo Move-To-Front è 2-competitivo quando il numero di operazioni sulla lista di n elementi è $O(n^2)$.

Esercizio 5

[punti 7]

Siano date K sequenze di interi, dove ciascuna sequenza contiene N interi distinti in ordine crescente ed è contenuta nella memoria esterna.

Si vuole calcolare l'intersezione delle K sequenze, definita come la sequenza crescente composta dagli interi che appaiono in tutte le K sequenze. (Tale operazione viene utilizzata nei motori di ricerca per rispondere alle query con i termini messi in AND.)

a) Progettare un algoritmo efficiente per effettuare tale intersezione ipotizzando che la memoria principale possa contenere $K+1$ blocchi di memoria e $O(K)$ celle aggiuntive di memoria. In altre parole, utilizzando il modello di memoria secondaria, abbiamo che la dimensione di un blocco è di B elementi e quindi la memoria può contenere $M = (K+1)B + O(K)$ elementi.

b) Analizzare il costo della soluzione proposta sia in termini del numero di operazioni di trasferimento I/O che di operazioni nella CPU (quest'ultimo è il classico costo del modello RAM).