

[Dashboard](#) / [I miei corsi](#) / [michele.ceccarelli@unina.it](#) / [Materiale Didattico Programmazione I](#) / [Esame 17-9-2021](#)

Domanda **1**

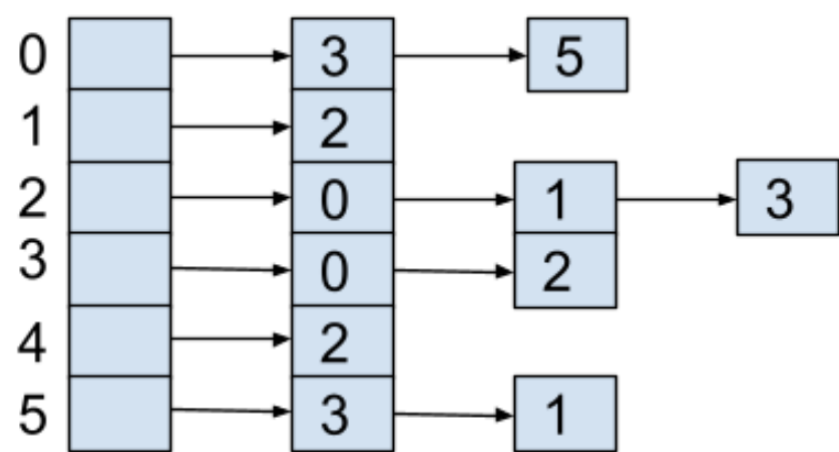
Risposta non ancora data

Punteggio max.: 1,00

Una nuova start-up vuole lanciare sul mercato un motore di ricerca basato su un fattore di peso (rank) da dare a ciascuna pagina in base alla sua importanza nella rete. Per ogni pagina si memorizzano:

- Un ID della pagina (potrebbe essere un intero fra 0 e $N-1$, dove N è il numero di pagine del web)
- Il numero di link contenuti in questa pagina (i link uscenti alle altre pagine del web). Per la pagina i indichiamo questo valore con $L(i)$
- Un valore di peso P che rappresenta l'importanza della pagina. Indichiamo questo valore per la pagina i con $P(i)$. All'inizio tutte le pagine hanno lo stesso rank pari a $1/N$.

L'insieme delle pagine sono mantenute in un vettore. Un Web è un vettore di N liste in cui nella lista i -ma ci sono gli ID delle pagine linkate da questa pagina. Ad esempio la pagina 0 contiene due link uno alla pagina 3 e uno alla pagina 5, la pagina 1 contiene un link alla pagina 2, etc.:



Implementare le strutture dati per rappresentare l'insieme delle pagine e il Web.

Implementare la funzione *rankUpdate* che aggiorna i pesi delle pagine in maniera tale che ad ogni iterazione ciascuna pagina trasferisce parte del suo rank in maniera uniforme a tutte le pagine citate nella pagina (i link uscenti da questa pagina). Quindi il rango di ogni pagina viene aggiornato nel seguente modo:

$$P(i) = \frac{(1-\delta)}{N} + \delta \sum_{j \in M(i)} \frac{P(j)}{L(j)}$$

Dove $M(i)$ sono tutte le pagine del Web che contengono un link alla pagina i , e δ è un valore costante tra 0 e 1. Nota i link da una pagina a sé stessa non vanno considerati. Per l'esempio in figura il rank della pagina 2 viene aggiornato come:

$$P(2) = \frac{1-\delta}{5} + \delta \cdot \left(\frac{P(1)}{1} + \frac{P(3)}{2} + \frac{P(4)}{1} \right)$$

Suggerimento importante: non è necessario che per aggiornare una certa pagina i si debba calcolare esplicitamente l'insieme delle pagine che contengono il link alla pagina i.

Il vostro programma inizializza un Web e visualizza il rank di ogni pagina, invoca la funzione *rankUpdate*, e poi visualizza il rank aggiornato.

Facoltativo 1. Invocate *rankUpdate* iterativamente fino alla convergenza. Per verificare la convergenza potete calcolare la media delle differenze in valore assoluto dei valori di P fra una iterazione e la successiva. Visualizzare la differenza sullo schermo e fermare le iterazioni quando la differenza è minore di un certo valore (es. 0.01).

Facoltativo 2. Implementare una funzione *initWeb* che istanzia le strutture del Web in maniera random.

Consegnare una cartella in formato compresso del tipo CognomeNome.zip (prima il cognome e poi il nome senza spazi!). Il file zip contiene una cartella CognomeNome con il codice sorgente ed il relativo Makefile. Tutto il codice deve essere opportunamente documentato. Non consegnare programmi che non compilano correttamente.

Complimenti! Avete implementato una versione semplificata dell'algoritmo PageRank di Google.

↵

i ▼

B

I

≡

≡

🔗

🔗

🖼️



Dimensione massima per i file nuovi: 8MB, numero massimo di allegati: 3



File



[◀ Esame 16-7-2021](#)

Vai a...