



Politecnico di Milano
Informatica B [Mod 2]
Algoritmi, Strutture Dati e Calcolo Parallelo
Prof. Pier Luca Lanzi & Ing. Daniele Loiacono

COGNOME E NOME

MATRICOLA

20 Settembre 2010

Risolvere i seguenti esercizi, scrivendo le risposte ed eventuali tracce di soluzione negli spazi disponibili. È possibile usare il libro di testo. **Non è permesso usare nessun supporto contenente esercizi svolti (eserciziari, slide, ecc.). Non è possibile consegnare esercizi scritti in matita. Non consegnare altri fogli.**

Spazio riservato ai docenti

--	--	--	--	--

Esercizio 1. Si consideri una tabella hash di dimensione 11, inizialmente vuota, e si inseriscano i seguenti elementi C, A, L, C, I, O, P, U, L, I, T, O, utilizzando la funzione di hash $h(k) = k \bmod 11$ per la k-esima lettera dell'alfabeto (per A, $k=1$, per B, $k=2$, ecc.) e probing lineare a passo uno. Si indichi il contenuto della tabella dopo aver cancellato nell'ordine G, A, L, L, I, A, N, I e poi nell'ordine R, I, V, E, R, A.

Esercizio 2. Dato un albero binario la visita in preordine restituisce A, E, B, F, G, C, D, I, H, la visita in postordine restituisce B, G, C, F, E, H, I, D, A mentre la visita in-ordine restituisce B, E, G, F, C, A, D, H. Si ricostruisca l'albero binario.

Esercizio 3. Si delinei, utilizzando pseudocodice o linguaggio C++, un algoritmo (di minima complessità asintotica) in grado di calcolare l'elemento mediano pesato inferiore di un vettore V di n interi v_i a cui siano associati n pesi w_i la cui somma è 1. Il mediano pesato inferiore è definito come quel valore v_m tale che,

$$\sum_{v_i < v_m} w_i < 1/2$$

$$\sum_{v_i > v_m} w_i \leq 1/2$$

Calcolate la complessità dell'algoritmo.

Esercizio 4: Il *Game of Life* è una simulazione che si svolge su una griglia celle. Ogni cella ha 8 vicini, che sono le celle ad essa adiacenti, includendo quelle in senso diagonale. Ogni cella può trovarsi in due stati: viva o morta (1 o 0). Ad ogni iterazione, lo stato della griglia evolve e tutte le celle del mondo vengono aggiornate simultaneamente secondo queste regole:

- una cella morta con esattamente 3 vicini vivi nasce, diventando viva.
- una cella viva con 2 o 3 vicini vivi sopravvive;
- altrimenti muore (per isolamento o sovraffollamento)

Implementare, utilizzando i pthread, un programma che aggiorna la griglia per n iterazioni, dove n viene letto da tastiera all'inizio del programma. Nell'implementare la soluzione non occorre leggere la configurazione iniziale della griglia, di dimensione $M \times M$, che si può assumere già inizializzata. Inoltre si implementi una soluzione pensata per un sistema di K processori (con K divisore di M).

Esercizio 5. Si consideri un problema di ottimizzazione discreto le cui soluzioni sono definite come una permutazione dei primi 10 caratteri dell'alfabeto, $\{a,b,c,d,e,f,g,h,i,l\}$. Per applicare il tabu-search a questo problema, l'intorno della soluzione corrente viene definito come tutte quelle soluzioni che si possono ottenere scambiando fra di loro due posizioni della permutazione corrente. Ad esempio,

a,b,c,d,e,f,g,h,i,l $\xrightarrow{\text{scambio (2,3)}}$ a,c,b,d,e,f,g,h,i,l

Sia la tabu-list la lista delle ultime N mosse effettuate, che non possono essere ripetute.

Dire se questa definizione di tabu-list garantisce di non ritornare alla medesima soluzione in N o meno mosse? Giustificare la risposta data.

Esercizio 4: Si consideri un vettore non ordinato A contenente n elementi e l'algoritmo per la ricerca di un elemento x all'interno di A che esamina gli elementi di A dal primo fino all'ultimo, sequenzialmente, controllando se $A[i]=x$ e, se x non è presente in A , termina quando ha esaminato tutti gli elementi. Si considerino tre tipi di array $A1$, $A2$ e $A3$ tali che

- $A1$ per cui esiste un unico indice i tale che $A[i]=x$
- $A2$ per cui non esiste un indice i tale che $A[i]=x$
- $A3$ per cui esistono esattamente $n/2$ indici i tali che $A[i]=x$

- 1) Calcolare la complessità nel caso pessimo per l'algoritmo di ricerca quando questo viene applicato a vettori di tipo $A1$, $A2$, e $A3$.
- 2)