

Informatica

Prova Scritta - 02 Settembre 2013

1. Si vuole calcolare l'autovalore massimo λ della matrice A di dimensione n utilizzando il metodo delle potenze:

$$\begin{cases} \mathbf{y}_{k+1} = A\mathbf{z}_k \\ \lambda_{k+1} = \|\mathbf{y}_{k+1}\|_2 \\ \mathbf{z}_{k+1} = \frac{1}{\lambda_{k+1}}\mathbf{y}_{k+1} \end{cases}$$

in cui $\|\cdot\|_2$ rappresenta la norma euclidea di un vettore. Scrivere un programma in FORTRAN 90 che:

- legge da un file esterno 'Dati.in' la dimensione n , la matrice A , il vettore iniziale \mathbf{z}_0 , la tolleranza di uscita TOLL e il numero massimo di iterazioni ITMAX allocando la memoria strettamente necessaria;
- Esegue i tre passi del metodo delle potenze fino a che non riduce il residuo:

$$\frac{\|\mathbf{y}_{k+1} - \lambda_{k+1}\mathbf{z}_k\|_2}{\|\mathbf{z}_k\|_2}$$

al di sotto della tolleranza o non supera il numero massimo di iterazioni;

- stampa l'autovalore massimo e il numero di iterazioni effettuate;

Il programma utilizzerà la subroutine MATVET per il prodotto matrice vettore e la function EUCL per il calcolo della norma euclidea. (Scrivere anche subroutine e function!)

2. Scrivere una modulo in FORTRAN 90 per la gestione degli esami universitari utilizzando la struttura dati della lista. Una lista rappresenterà un singolo esame e ogni nodo di questa lista conterrà a sua volta una variabile di tipo studente composta da:

- Nome dello studente in una variabile di tipo *character* lunga 100;
- Numero di matricola dello studente in una variabile di tipo *integer*;
- Esito dell'esame in una variabile di tipo *integer*;

Scrivere:

- (a) Una subroutine *Inserisci_Studente* per l'inserimento di un nuovo studente (nome e numero di matricola) nella lista d'esame con inizializzazione del voto a 0 (valore che indica che l'esame deve ancora essere provato);
 - (b) Una function *Agg_Voto* per aggiornare il voto d'esame di uno studente che corrisponde ad un determinato numero di matricola. Questa function ritorna 0 se lo studente non ha ancora sostenuto l'esame, 1 se lo studente ha già un voto e -1 se lo studente non compare nella lista;
3. Con riferimento all'esercizio precedente, studiare la complessità della function *Agg_Voto* e dire se esiste una struttura dati alternativa con cui è possibile ridurre la complessità. In caso affermativo descrivere brevemente questa seconda opzione e la sua complessità.

4. Descrivere brevemente le strutture dati (statiche e dinamiche) che si possono utilizzare per rappresentare gli alberi e la loro implementazione in FORTRAN 90 (solo la struttura dati).
5. Descrivere brevemente la memorizzazione dei numeri reali in singola precisione secondo lo standard IEEE 854 e scrivere la rappresentazione del numero -82.125.
6. (*facoltativo*) Scrivere una subroutine per il modulo dell'esercizio 2 che permetta di confrontare due liste di esami ed estrarre nome, numero di matricola e voto degli studenti che hanno superato entrambi gli esami. Questa subroutine riceverà in input due variabili di tipo derivato *Esame_1* ed *Esame_2* e stamperà a video numero di matricola, nome studente e voto solo per gli studenti che hanno ottenuto almeno 18.

Tempo: 2 ore 30 minuti. (Punteggi: 10,10,3,4,3,5).