

Esercizi sulle tecniche di ricerca

(Fondamenti di Informatica 2 – Walter Didimo)

Esercizio 1 Scrivere un metodo di classe di nome *cerca* che prende in ingresso un array *seq* di stringe già ordinate alfabeticamente in senso non decrescente, e una ulteriore stringa *elem*, e che restituisce:

- la posizione di *elem* in *seq*, se *elem* è contenuta in *seq*
- -1, se *elem* non è contenuta in *seq*

Il metodo deve essere implementato con una tecnica di ricerca dicotomica, in modo che la sua complessità computazionale sia $O(\log N)$, dove N è la dimensione dell'array *seq*.

Scrivere inoltre un metodo *main* che effettua il test del metodo *cerca*. In particolare, il metodo *main* deve far inserire all'utente una sequenza di stringhe, assicurandosi che tale sequenza sia ordinata alfabeticamente; farà poi inserire una ulteriore stringa da ricercare e visualizzerà all'utente l'esito della ricerca.

Esercizio 2 Un oggetto della classe *StringheOrdinate* rappresenta una sequenza di stringhe ordinate alfabeticamente in senso non decrescente. La sequenza può contenere un numero massimo di stringhe predefinito in fase di creazione dell'oggetto. Lo scheletro della classe è come segue:

```
class StringheOrdinate{

    private String[] seq; // sequenza ordinata di stringhe
    private int count;    // numero di elementi memorizzati nell'array

    /* crea un'istanza della classe, fissando a dim la dimensione massima
       della sequenza */
    public StringheOrdinate (int dim){...}

    /* aggiunge l'elemento elem se c'è ancora spazio nell'array; l'elemento
       viene inserito nella posizione giusta, mantenendo cioè l'ordinamento
       alfabetico. In caso di inserimento, il metodo restituisce la
       posizione del nuovo elemento, altrimenti restituisce -1 */
    public int aggiungi (String elem){...}

    /* ricerca l'elemento elem; se presente, ne restituisce la posizione,
       altrimenti restituisce -1 */
    public int cerca (String elem){...}

    /* fornisce una descrizione completa della sequenza di elementi */
    public String toString (){...}
}
```

Svolgere i seguenti punti:

1. Implementare la classe *StringheOrdinate*;
2. Analizzare la complessità asintotica di ciascun metodo implementato.
3. Scrivere una classe di test che verifica il corretto funzionamento della classe *StringheOrdinate*.

Esercizio 3 Ampliare la classe `StringheOrdinate` dell'esercizio precedente, aggiungendo il seguente metodo di istanza:

```
/* elimina l'elemento elem se presente. In caso di eliminazione
restituisce true; se l'elemento non è presente, restituisce false. */
public boolean elimina (String elem){...}
```

Esercizio 4 Un oggetto della classe `RubricaTelefonica` rappresenta una rubrica che permette di memorizzare, e ricercare utenze telefoniche in numero qualunque. Lo scheletro della classe è come segue. Una utenza telefonica è modellata attraverso la seguente classe:

```
class Utenza{
    private String cognome, nome, telefono;

    /* crea una utenza con i dati specificati */
    public Utenza (String cognome, String nome, String telefono){..}

    /* restituisce il cognome dell'utenza */
    public String getCognome () {..}

    /* restituisce il nome dell'utenza */
    public String getNome () {..}

    /* restituisce il telefono dell'utenza */
    public String getTelefono () {..}

    /* restituisce una descrizione completa dell'utenza */
    public String toString (){..}
}
```

La classe `RubricaTelefonica` ha il seguente scheletro:

```
class RubricaTelefonica{
    private Utenza[] seq; // memorizza la lista delle utenze
    private int count; // memorizza il numero di utenze correnti

    /* crea una rubrica telefonica vuota */
    public RubricaTelefonica (){..}

    /* aggiunge alla rubrica l'utenza specificata, mantenendo le utenze
ordinate alfabeticamente per cognome */
    public void aggiungi (String cognome, String nome, String tel){..}

    /* restituisce la descrizione completa dell'utenza con cognome e
nome specificati; se l'utenza non esiste, restituisce null */
    public String cerca(String cognome, String nome){..}

    /* restituisce una descrizione completa della rubrica */
    public String toString (){..}
}
```

Si chiede di: (i) implementare le classi `Utenza` e `RubricaTelefonica`; (ii) implementare una classe di test per la classe `RubricaTelefonica`. Si noti in particolare che la rubrica deve poter contenere un numero arbitrariamente elevato di utenze. Questo implica che se il metodo `aggiungi` non trova spazio per inserire una nuova utenza perché l'array `seq` è pieno, il metodo dovrà preventivamente sostituire `seq` con un array di dimensione maggiore e copiarci dentro gli elementi già presenti. Una scelta tipica è quella di sostituire `seq` con un array di dimensione doppia.