FI2 (12 CFU) - Compito di esame del 21 giugno 2011 – a.a. 2010-11, *tempo a disposizione: 90 minuti*[1]

Problema 1

Si considerino i seguenti metodi:

```
public class Complessita_F2{
  public static boolean[] append(boolean[] vet, boolean b){
    boolean[] newVet=new boolean[vet.length+1];
    for (int i=0;i<vet.length;i++){
      newVet[i]=vet[i];
    }
    newVet[vet.length]=b;
    return newVet;
  }

  public static boolean[] bit(int n){
    if (n==0){
      return new boolean[0];
    }
    boolean b = n%2!=0 ;
    return append( bit(n/2) , b );
  }
}
```

(a) Calcolare, motivandolo adeguatamente, il costo computazionale del metodo `append` in funzione della dimensione dell'input.

(b) Calcolare, motivandolo adeguatamente, il costo computazionale del metodo `bit` in funzione del valore di `n` e della dimensione dell'input.

(c) Riscrivere il metodo `bit` senza utilizzare la ricorsione.

Problema 2

Con riferimento agli alberi generici contenente etichette **di tipo carattere** si richiede quanto segue:

(a) Definire il TDA albero generico elencando i metodi principali che lo caratterizzano. Realizzare poi le classi `Tree` e `TreeNode` che rappresentano rispettivamente un albero generico e un suo nodo (solo variabili di istanza e firma dei metodi pubblici del TDA e costruttori).

(b) Realizzare il metodo statico la cui firma è `boolean startWith(TreeNode root, String prefix)` che restituisce `true` se esiste almeno un percorso da `root` che attraversa nodi le cui etichette, concatenate insieme, permettono di ottenere la stringa `prefix`. *Il metodo dovrà avere costo lineare in funzione del numero di nodi dell'albero.*

(c) Realizzare il metodo statico la cui firma è `void remove(String s)` che rimuove dall'albero `this` tutti i nodi (e relativi sottoalberi) i cui caratteri compaiono in `s`. Discutere il costo computazionale dell'algoritmo.

Problema 3

Si vuole rappresentare un grafo stradale. I nodi (che indicano gli incroci tra strade) sono descritti da un valore intero (tra 0 e N-1). Gli archi orientati (che rappresentano i tratti di strada che uniscono gli incroci) sono caratterizzati dalle seguenti informazioni: *tempo di percorrenza* (in minuti), *denominazione* (nome del tratto di strada). Si richiede quanto segue:

(a) Realizzare la classe `RouteGraph` (e le eventuali classi relative ad archi e nodi) che rappresenta il grafo stradale mediante liste di adiacenza. Le classi dovranno contenere, oltre alle variabili di istanza, anche le firme di tutti i metodi pubblici che si ritiene fondamentali per risolvere i problemi successivi.

(b) Realizzare un metodo della classe `RouteGraph` la cui firma è `List<Integer> isocrone(int source, int min, int max)` che restituisce la lista di tutti i nodi che sono raggiungibili dal nodo `source` percorrendo il tragitto più rapido, in un tempo (in minuti) compreso tra `min` e `max`. Descrivere poi il costo computazionale dell'algoritmo, motivandolo opportunamente.

(c) Realizzare un metodo della classe `RouteGraph` la cui firma è `void streetCleaner()` che rimuove dal grafo `this` tutti gli archi che non hanno la denominazione della via (stringa nulla). Descrivere poi il costo computazionale dell'algoritmo, motivandolo opportunamente.

[1] Compilare i campi nel retro del foglio e seguire le istruzioni indicate per la consegna.

Consegna dell'elaborato

1) Compilare i campi con le proprie informazioni

2) Piegare lungo la linea verticale ed utilizzare il foglio a meta' per contenere l'elaborato.

(Questa parte va dietro)

Compito d'esame - Fondamenti di Informatica II (FI2)

Matricola

Nominativo

Esame a cui si partecipa
Fondamenti di Informatica II