Esercitazione 8

29 dicembre 2015

Termine per la consegna dei lavori: **martedì 12 Gennaio 2016** ore **23.59** (due settimane di tempo).

Istruzioni

I lavori dovranno essere salvati in una cartella che deve contenere tutto e solo ciò che volete venga consegnato e valutato, dovrà essere **obbligatoriamente** un file **eseguibile** con estensione `.py` per **ognuno** degli esercizi. Controllate che l'esecuzione del comando:

```
python <nome_file>.py
```

per ognuno degli esercizi produca l'output desiderato.
Per consegnare gli elaborati dovete raggiungere la cartella contenente i file da inviare in modalità terminale (`cd path_della_cartella`) e quindi eseguire il comando:

```
consegna consegna8
```

verrà visualizzata la lista di tutto ciò che è stato inviato.
Consegne successive (entro il termine per la consegna) sovrascriveranno le precedenti, verrà valutata solo l'ultima consegna sottomessa.
È obbligatorio che all'interno di ogni file sia riportato il vostro nome, cognome e numero di matricola (riportati all'interno di una riga commento all'inizio del file, es: `#Mario Rossi 1234567`).

ATTENZIONE!

Gli unici moduli importabili ammessi in questa esercitazione sono i moduli `math` e `random`. Esercizi risolti utilizzando altri moduli importati riceveranno il punteggio minimo. Python contiene delle **built-in functions**, che potete utilizzare e la cui lista si può trovare a questo indirizzo: `https://docs.python.org/2/library/functions.html`.

Esercizio 1

Definire una classe `Polinomio` che rappresenta un polinomio di grado n

$$c_n x^n + c_{n-1} x^{n-1} + ... + c_1 x + c_0.$$

In particolare la classe deve contenere:

1. un costruttore (`__init__`) con un parametro `coeff` che come valore di default vale `[0]` (polinomio nullo) e in generale è la lista dei coefficienti del polinomio in ordine crescente `[c_0, c_1, ..., c_n]`, ad esempio deve essere possibile istanziare un polinomio con l'istruzione `p = Polinomio([4, 4, 1])`;

2. la definizione del metodo speciale `__repr__(self)` che ritorna una rappresentazione per il polinomio, ad esempio quella sottostante;

   ```
   x^ 2 + 4 x + 4
   ```

3. un metodo `valuta(self, x)` che, dato un reale `x`, restituisce il valore del polinomio valutato in `x`.

4. un metodo `derivata(self)` che stampa la derivata del polinomio.

5. la definizione dei metodi speciali:

   - `__add__(self, p)` che, dato il polinomio `p`, restituisce il polinomio somma dei due.
   - `__sub__(self, p)` che, dato il polinomio `p`, restituisce il polinomio differenza dei due.
   - `__mul__(self, p)` che, dato il polinomio `p`, restituisce il polinomio prodotto dei due.
   - `diviso(self, p)` che, dato il polinomio `p`, stampi i polinomi quoziente e resto della divisione per `p`.

Per ognuno di questi metodi dare almeno un esempio di invocazione.
Nota: i metodi `__add__`, `__sub__` e `__mul__` vengono invocati, rispettivamente, con gli operatori `+`, `-` e `*`. Mentre l'operatore `__repr__` è invocato quando un oggetto di tipo `Polinomio` viene stampato con `print`.

Esercizio 2

Un albero binario T è chiamato *albero di Fibonacci* se è un albero vuoto, oppure se per ogni nodo n di T vale la seguente proprietà: n è una foglia oppure le altezze dei sottoalberi destro e sinistro di n differiscono *esattamente* di 1.
Definire la funzione `isFibonacci(T)` che restituisca `True` o `False` a seconda che `T` sia o meno un albero di Fibonacci. Si utilizzi la classe `Albero` costruita in laboratorio e tutte le funzioni di utilità che possono esservi utili.
Riportare almeno due esempi di invocazione, uno su un albero di Fibonacci non vuoto e uno che ritorni `False`.
`http://it.wikipedia.org/wiki/Albero_di_Fibonacci`

Esercizio 3 - Animal Game

Animal Game è un gioco per computer che utilizza un albero binario per simulare il gioco di *Indovina l'animale*.

All'inizio della partita il giocatore pensa ad un animale, con l'obiettivo di trovarne uno che il computer non riesca ad indovinare.

I nodi dell'albero sono costituiti da domande alle quali l'utente deve rispondere per permettere al computer di indovinare; le foglie invece sono costituite dagli animali che il computer conosce.

Il programma è strutturato come segue:

- partendo dalla radice dell'albero, vengono poste le domande all'utente e a seconda della risposta ci si muove a destra o a sinistra lungo l'albero;
- raggiunta una foglia, viene ipotizzato l'animale;
- se la risposta non è corretta viene chiesto all'utente di inserire una nuova domande in modo da poter distinguere l'animale ipotizzato da quello corretto.

Un esempio di gioco è il seguente:

```
Pensa ad un animale. Rispondi S o N alle domande.

Mammifero? S
Nitrisce? N
La risposta e' "Giraffa"? N
Hai vinto! A quale animale stavi pensando? Gatto
Inserisci una domanda per distinguere "Gatto" da "Giraffa": Ha un collo
    lungo?
Quale sarebbe la risposta per "Gatto"? N

Vuoi giocare ancora? S
Pensa ad un animale. Rispondi S o N alle domande.

Mammifero? S
Nitrisce? N
Ha un collo lungo? N
La risposta e' "Gatto"? S
Ho indovinato!
```

`https://it.wikipedia.org/wiki/Animal_(videogioco)`

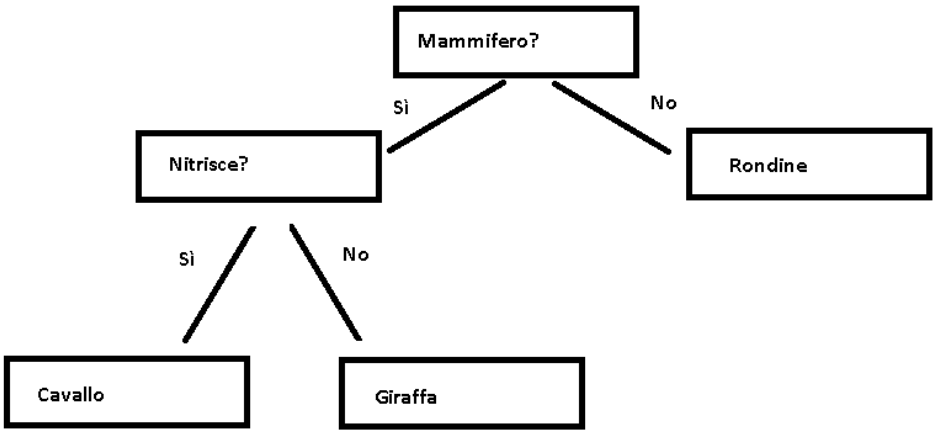

Figura 1: Albero Binario degli Animali PRIMA dell'aggiunta della nuova domanda

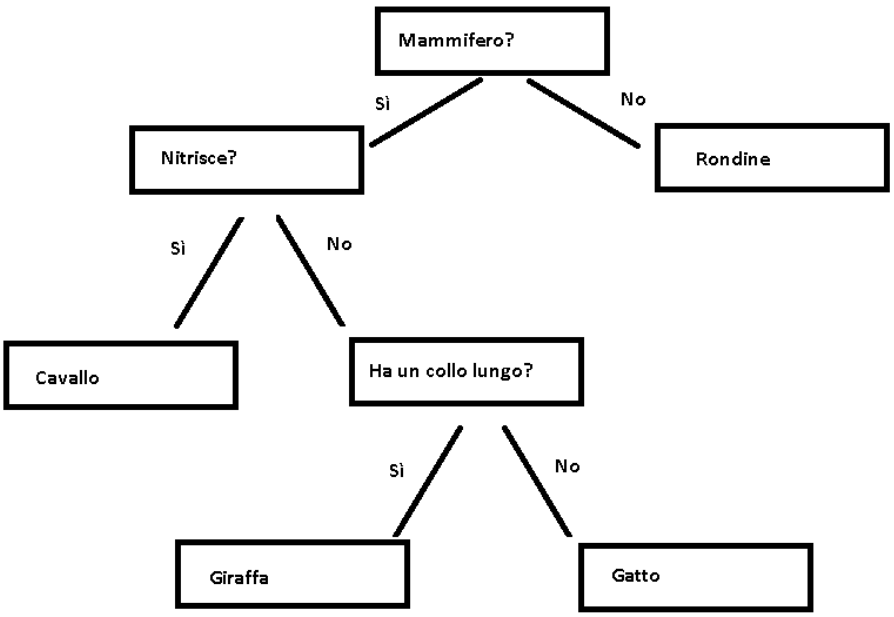

Figura 2: Albero Binario degli Animali DOPO l'aggiunta della nuova domanda