

Lab Lesson 07

Giacomo Bergami

March 6, 2018

Esercizi

Per controllare se effettivamente avete svolto bene gli esercizi, questa volta fornisco anche dell'ulteriore codice, che controlla la correttezza delle funzioni da voi implementate. Trovate nei file readme maggiori informazioni su come gestire il codice e la valutazione.

0. Scrivere un programma ricorsivo che dica se una stringa è palindroma. **Suggerimento:**
 - Quando non devo applicare la ricorsione? Che cosa restituisco? (*questi sono detti in gergo **casi base***).
 - Quanto invece devo applicare la ricorsione? (*questa è detto **casi induttivo***)
Su quale sottostringa applico la ricorsione? Come combino la valutazione del caso corrente con il risultato della ricorsione?
1. Scrivere un programma ricorsivo, che valuti una espressione aritmetica. Questa espressione aritmetica può essere costituita solamente da numeri interi, con o senza segno, e le uniche operazioni aritmetiche previste sono la somma, la sottrazione e la moltiplicazione. **Suggerimento:**, per risolvere problemi complessi, occorre prima spezzettare i sottoproblemi:
 - Metodo **signReduction**. Data una stringa **signExpr** contenente i segni + o -, implementare su di essa la regola dei segni. Restituire 1 se il segno finale è positivo, e -1 altrimenti. **Non è richiesta la ricorsione.**
 - Metodo **rightInteger**. Scandire verso destra una stringa **str** a partire dalla posizione **start**, e restituire l'indice intero **pos** che definisce un numero compreso tra **start** e **pos**.
 - Metodo **signedNumber**. Dato un numero intero preceduto da una serie di segni algebrici + o -, restituire la rappresentazione intera corretta. **Può richiedere l'uso di **rightInteger**.**
 - Effettuare la somma algebrica di un'espressione, contenente numeri con segno e solo somme e sottrazioni. **Usare la ricorsione. Quali due metodi già implementati posso usare?**

- Metodo **checkParentheses**. Data una stringa **s**, restituisce un oggetto di classe **PairOfIntegers**. Questo oggetto fornisce una coppia d'interi $\langle left, right \rangle$, che indica che la stringa **s[left]...s[right-1]** è il contenuto di tale parentesi bilanciata. Ricorda come funziona **substring** in Java. Questa implementazione semplificherà i passaggi successivi.
- Metodo **leftInteger**. Scandire verso sinistra una stringa **str** a partire dalla posizione **start**, e restituire l'indice intero **pos** che definisce un numero (comprensivo di segno, solamente negativo) compreso tra **pos** e **start**.
- Metodo **evaluate**. Vi ricordo le regole dell'aritmetica:
 - Se è un solo numero...
 - Prima si valutano le sottoespressioni tra parentesi. Per ogni coppia di parentesi bilanciate, procedere ricorsivamente finché non ritroviamo la coppia di parentesi più annidata (Quale metodo già implementato ricorda?). Sostituire la sottoespressione al valore valutato, e...
 - All'interno di ciascuna parentesi (che non ne contiene altre) o in un'espressione senza parentesi, prima effettuare tutti i prodotti e poi effettuare le somme e le sottrazioni rimaste. Quest'ultima parte, quale metodo già implementato ricorda?