

Esercitazioni di Algoritmi e Strutture Dati



Il esercitazione, 14/03/2016

Tong Liu

SOLUZIONE DEL COMPITO PRECEDENTE

Definire e inizializzare un array di N numeri interi *distinti* (la scelta di N e degli elementi dell'array è arbitraria).

Definire una funzione `search` che, presi in input un array A e un numero intero x , restituisca:

- i , se l' $(i+1)$ -esimo elemento di A è x (cioè, $A[i] == x$);
- -1 , se l'elemento x non occorre in A (cioè, per ogni $0 \leq i < N$, $A[i] \neq x$).

OGGI

- Ripassare i concetti di Coda, Pila, Liste
- Esercizi vari

CODA (QUEUE)

- Coda (Queue) FIFO
- Dati: una sequenza S di n elementi
- Operazioni:
 - `isEmpty()` -> result, restituisce true se S e' vuota e false altrimenti
 - `enqueue(elem e)` aggiunge e come ultimo elemento di S
 - `dequeue()` -> elem, toglie da S il primo elemento e lo restituisce
 - `first()` -> elem, restituisce il primo elemento di S (senza toglierlo da S)

PILA (STACK)

- PILA (Stack) LIFO
- Dati: una sequenza S di n elementi
- Operazioni:
 - `isEmpty()` -> result, restituisce true se S e' vuota e false altrimenti
 - `push(elem e)` aggiunge e come primo elemento di S
 - `pop()` -> elem, toglie da S il primo elemento e lo restituisce
 - `top()` -> elem, restituisce il primo elemento di S (senza toglierlo da S)

PILA (STACK) : NOTAZIONE POLACCA INVERSA

- Reverse Polish Notation(RPN)
- Sintassi utilizzata per le formule matematiche ed è dovuta a Jan Lukasiewicz, 1958.
- Vantaggio: eliminare le problematiche causate dalle parentesi e dalla precedenza degli operatori; risparmiare la memoria.
- Espressione: prima gli operandi poi gli operatori. Esempio, $3+2 \Rightarrow 3\ 2\ +$,
 $10 / 5 \Rightarrow 10\ 5\ /$

PILA (STACK) : NOTAZIONE POLACCA INVERSA

- Procedura:
 - usare una pila per accumulare gli operandi. Un operatore preleva dalla cima della pila tutti gli operandi che ha bisogno, esegue l'operazione e rimette in pila il risultato.
 - L'elemento più in basso è da considerarsi sempre l'operando sinistro.
 - Dopo tutte le operazioni sulla pila si avrà il risultato finale
- esempio:
 - $5 + (10 * 2) \Rightarrow 5 \ 10 \ 2 \ * \ +$
 - $((10 * 2) + (4 - 5)) / 2 \Rightarrow 10 \ 2 \ * \ 4 \ 5 \ - \ + \ 2$

NOTAZIONE POLACCA INVERSA

Vediamo ora una sequenza di slide con un esempio di calcolo con RPN:

L'espressione tradizionale da calcolare è $(4+3)*(5-2)=21$

L'espressione tradotta in RPN da calcolare è $4\ 3\ +\ 5\ 2\ -\ *$, ovvero :

4	3	+	5	2	-	*
---	---	---	---	---	---	---



STAC



In questo stato lo
stack è vuoto



STACK POINTER

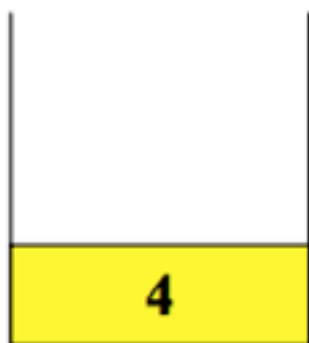
NOTAZIONE POLACCA INVERSA

4	3	+	5	2	-	*
---	---	---	---	---	---	---



Push (4)

STAC



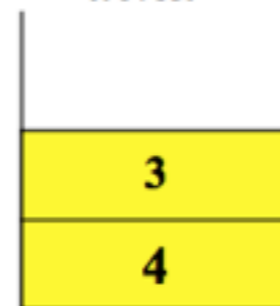
STACK POINTER

4	3	+	5	2	-	*
---	---	---	---	---	---	---



Push (3)

STAC



STACK POINTER

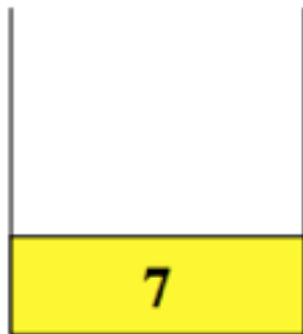
NOTAZIONE POLACCA INVERSA

4	3	+	5	2	-	*
---	---	---	---	---	---	---



Pop(3)
Pop(4)
4+3
Push(7)

STAC



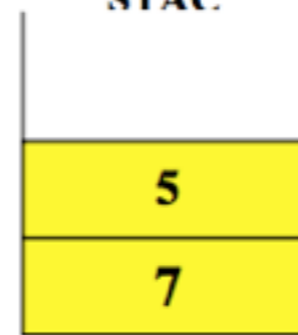
STACK POIN

4	3	+	5	2	-	*
---	---	---	---	---	---	---



Push (5)

STAC



STACK POINTER

NOTAZIONE POLACCA INVERSA

4	3	+	5	2	-	*
---	---	---	---	---	---	---



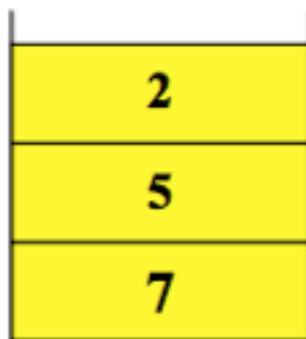
Push(2)

4	3	+	5	2	-	*
---	---	---	---	---	---	---



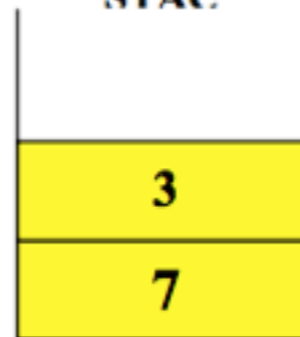
Pop (2)
Pop(5)
5-2
Push(3)

STAC



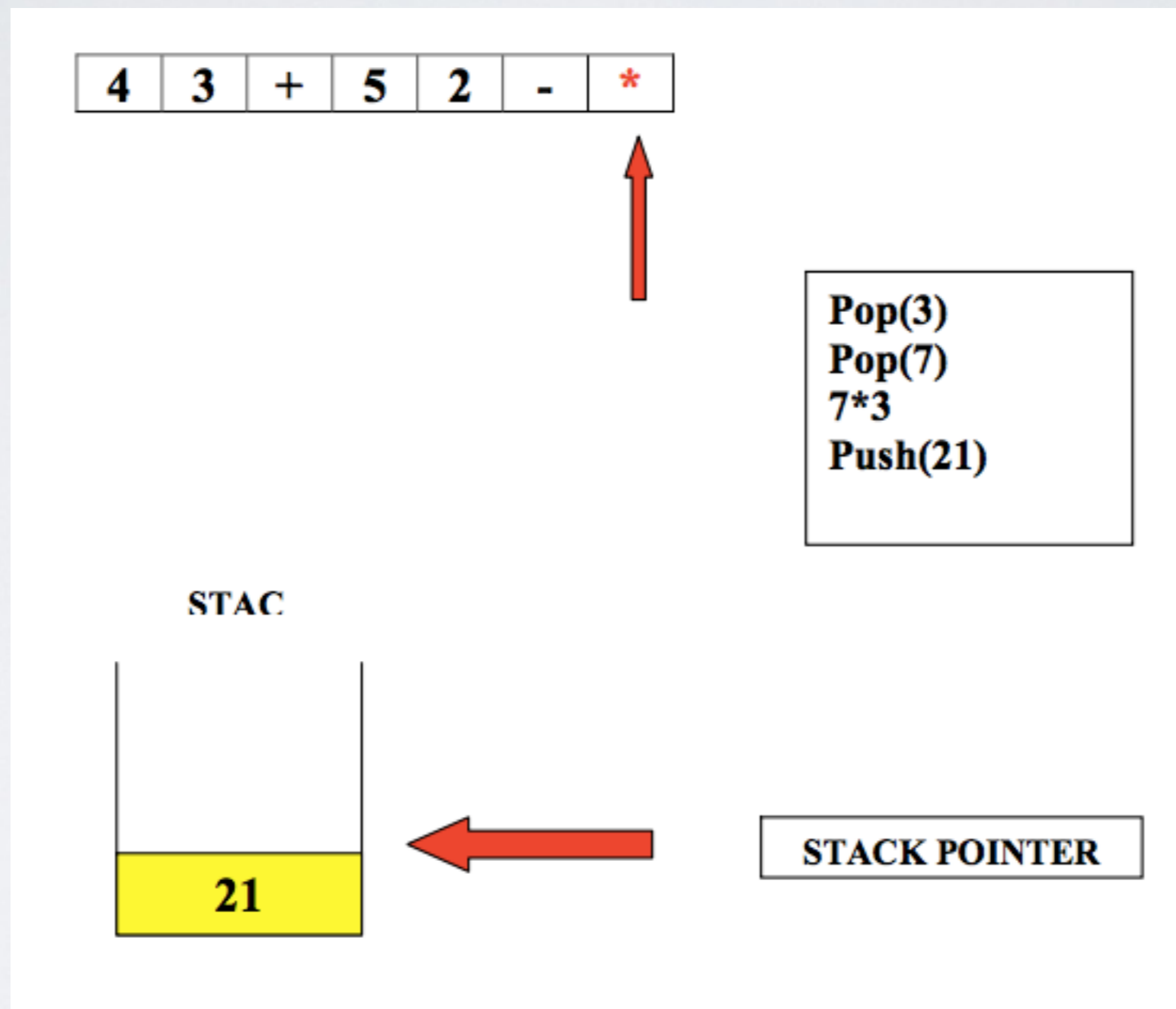
STACK POINTER

STAC



STACK POINTER

NOTAZIONE POLACCA INVERSA



LISTE

- LISTE - Liste con puntatori (Linked lists)
- Basato sull'uso di puntatori, no limiti alla dimensione della sequenza.
- Operazioni:
 - `empty()`, restituisce true se la lista non contiene oggetti, false altrimenti
 - `Pos head()`, restituisce la posizione del primo elemento
 - `Pos next(Pos p)`, restituisce la posizione dell'elemento successivo al p
 - `finished(Pos p)`, restituisce true se si e' oltrepassato un estremo della lista

LISTE

- `Item read(Pos p)`, legge il valore di un elemento della posizione p
- `write(Pos p, Item v)`, sovrascrive il valore dell'elemento p con il valore v
- `insert(Pos p, Item v)`, inserire un elemento nella posizione p della lista
- `remove(Pos p)`, rimuovere l'elemento nella posizione p della lista

ESERCIZI

ESERCIZIO SU CODA

- Es2.1 [Libro 3.7 variante] Si scriva una procedura che, data una coda L di interi genera un'altra coda L' in modo che tutti gli elementi dispari precedano, nello stesso ordine che avevano inizialmente in L , tutti gli elementi pari (per esempio, se $L = 3, 7, 8, 1, 4$, allora si ottiene $L' = 3, 7, 1, 8, 4$),

ESERCIZIO SU PILA

- Es 2.2. Un compito comune del compilatore e l'editore codice è determinare se le parentesi sono bilanciate e nestate bene, per esempio la stringa, `((()())())` le parentesi sono bilanciate, invece `)()()` e `()` non sono. Proporre un algoritmo che adopera la struttura pila e che ritorna true se le parentesi sono bilanciate e false altrimenti, se possibile, ritorna anche la posizione della parentesi che ha violato il bilanciamento. Input un array di char.

ESERCIZI SU LISTE

- Es 2.3, Esame Febbraio 2014: Data una lista L di interi si vuole togliere da L ogni elemento pari e inserirlo in una nuova lista M , mantenendo in entrambe le liste l'ordine originario degli elementi.

ESERCIZI SU LISTE

- Es 2.4, Esame Gennaio 2014: Data una lista L di interi si vuole modificarla cancellando tutti gli elementi adiacenti che hanno valori consecutivi mantenendo l'ordine iniziale degli elementi (p.e. se l'ingresso è 4,5,6,2,7,3,2,1,2,5 allora il risultato è 2,7,5).

ESERCIZI SU LISTE

- Es 2.5, Il rango di un elemento di una lista di interi é la somma degli elementi successivi piú se stesso.
- Rango di $[3,2,5]$ é $[10,7,5]$
- Creare una funzione che calcola il rango di una lista

ALCUNI RIFERIMENTI

- Camil Demetrescu, Irene Fionocchi, Giuseppe F, italiano, Algoritmi e strutture dati 2/ed, McGraw-Hill, 2008, ISBN 978 88 386 64687
- Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2009) [1990]. Introduction to Algorithms (3rd ed.). MIT Press and McGraw-Hill. ISBN 0-262-03384-4
- <http://www.algorithm.cs.sunysb.edu/algowiki/index.php/Data-structures-TADM2E-2>
- <http://www.science.unitn.it/~valter/Didattica/LabProg/risorse/manuale.pdf>