Programmazione e Algoritmi 1

A.A. 2023/24 — Soluzioni compito del 12 febbraio 2024

prof. Gianluca Amato

Esercizio 1 – 4

Controllare i file sorgente allegati a questa soluzione.

Esercizio 5 (5 punti)

L'algoritmo di ricerca lineare prende in input una lista `l` ed un valore `x`, e restituisce la posizione di `x` in `l`, oppure un valore speciale se `x` non è presente in `l`. Nel seguito assumeremo che questo valore speciale sia `-1`. L'algoritmo funziona scandendo la lista `l` dalla prima all'ultima posizione, fermandosi non appena viene trovato `x` e restituiendone la posizione. Se la scansione termina senza che `x` sia stato trovato, l'algoritmo restituisce `-1`.

Nel caso pessimo in cui l'elemento `x` non si trovi in `l`, l'algoritmo di ricerca lineare esegue un numero di operazioni proporzionale alla lunghezza n, della lista `l`, per cui la sua complessità è $O(n)$.

L'algoritmo di ricerca binaria ha stessi input e stesso valore di ritorno della ricerca lineare, ma ha come requisito aggiuntivo che la lista `l` sia ordinata. Grazie a questo requisito, l'algoritmo di ricerca binaria può eseguire la ricerca in modo più efficiente. L'algoritmo funziona in questo modo:

1. Se la lista è vuota, restituisce `-1`.
2. Altrimenti,
   (a) calcola l'indice `m` che corrisponde all'elemento centrale della lista.
   (b) Se `l[m]` è uguale a `x`, restituisce `m`.
   (c) Se `l[m]` è maggiore di `x`, esegue la ricerca binaria sulla sottolista che va dal primo elemento all'elemento in posizione `m-1`.
   (d) Se `l[m]` è minore di `x`, esegue la ricerca binaria sulla sottolista che va dall'elemento in posizione `m+1` all'ultimo.

Nel caso pessimo in cui l'elemento `x` non si trova in `l`, l'algoritmo di ricerca lineare esegue un numero di operazioni proporzionale al logaritmo della lunghezza n della lista `l`, per cui la sua complessità è $O(\log n)$. L'algoritmo è quindi più efficiente della ricerca lineare, ma ha lo svantaggio di richiedere che la lista `l` sia ordinata.

Ricerca lineare del numero 10 nella lista contenente i numeri da 1 a 100. L'algoritmo esegue i seguenti passi:

passo	posizione i sotto esame	valore in posizione i
1	0	99
2	1	48
⋮	⋮	⋮
10	9	10

Dunque al decimo passo l'algoritmo restituisce 9.

Ricerca binaria del numero 10 nella lista contenente i numeri da 1 a 100. L'algoritmo esegue i seguenti passi:

passo	pos. iniziale sottolista	pos. finale sottolista	indice centrale	valore centale
1	0	99	49	50
2	0	48	24	25
3	0	23	11	12
4	6	10	8	9
5	9	10	9	10

Dunque al quinto passo l'algoritmo restituisce 9.

Esercizio 6 (8 punti)

riga programma					valore variabili	note
10						call ackermann(1,2)
	1				m=1 n=2	
	2				m=1 n=2	
	4				m=1 n=2	
	7				m=1 n=2 x=?	call ackermann(1,1)
		1			m=1 n=1	
		2			m=1 n=1	
		4			m=1 n=1	
		7			m=1 n=1 x=?	call ackermann(1,0)
			1		m=1 n=0	
			2		m=1 n=0	
			4		m=1 n=0	
			5		m=1 n=0	call ackermann(0, 1)
				1	m=0 n=1	
				2	m=0 n=1	
				3	m=0 n=1	return 2
			5		m=1 n=0	return 2
		7			m=1 n=1 x=2	
		8			m=1 n=1 x=2	call ackermann(0, 2)
				1	m=0 n=2	
				2	m=0 n=2	
				3	m=0 n=2	return 3
		8			m=1 n=1 x=2	return 3
	7				m=1 n=2 x=3	
	8				m=1 n=2 x=3	call ackermann(0, 3)
		1			m=0 n=3	
		2			m=0 n=3	
		3			m=0 n=3	return 4
	8				m=1 n=2 x=3	return 4
10						print 4