

Programmazione e Algoritmica - B (1)
Terza prova in itinere, 20 Marzo 2024

Cognome Nome:

N. Matricola:

Esercizio 1 – Tabelle Hash

Si mostri la Tabella hash ottenuta inserendo le chiavi 14, 16, 12, 3, 1, 111 utilizzando un doppio hash $h(k,i) = (h_1(k) + i \cdot h_2(k)) \bmod m$ in una tabella di dimensione $m = 11$, dove $h_1(k) = k \bmod m$, e $h_2(k) = 1 + (k \bmod (m-1))$, indicando per ogni chiave k :

- a) $h_1(k)$ ogni volta che viene calcolato,
- b) $h_2(k)$ ogni volta che viene calcolato,
- c) la sequenza di ispezione
- d) i conflitti eventualmente generati.

Esercizio 2 – Semantica statica e dinamica delle funzioni

Dato il seguente frammento di codice L,

```
var z: Int = 7;
function test(var a: Int, const a: Bool) -> Int {
  var res: Int = a;
  if (a == true) res = res + z;
  return res;
}
var z: Int = 10;
print(test(20, true));
```

1. Si **dica** se il frammento di codice è corretto staticamente o no. Se non è corretto, si **dica** perché e si **corregga** l'errore
2. Applicando le regole (FS3) di semantica statica delle funzioni, si **costruisca l'ambiente statico** Δ_0 **associato ai parametri formali** della funzione test
3. Si **indichi** il contenuto dell'**ambiente statico** dopo la dichiarazione della funzione
4. In relazione **all'analisi dinamica**, si **indichi** il contenuto **dell'ambiente dinamico e della memoria** dopo la dichiarazione della funzione test nel caso di **scoping statico** e nel caso di **scoping dinamico** (**indicando** chiaramente la lambda- astrazione nei due casi)
5. Si **dica** infine se ci sono differenze fra lo scoping statico e lo scoping dinamico, **giustificando** la risposta (senza costruire il sistema di transizioni).

Esercizio 3 – Ricorsione e iterazione

Si consideri la seguente funzione ricorsiva, in cui u rappresenta il nodo di un **albero binario di ricerca**

```
Mistero (u) {  
    if (u == NIL) return NIL;  
    if (u.right != NIL) return Mistero(u.right);  
    else return u;  
}
```

1. Si **dica** quale problema risolve la funzione
2. Si **dica** se la funzione proposta è **ricorsiva in coda** o meno, **motivando** la risposta.
3. Si **scriva** una funzione iterativa che risolve lo stesso problema.

Esercizio 4 – Alberi Binari

- a) Si scriva lo pseudocodice di un algoritmo che, dati un albero binario (ovvero data la sua radice u) e una chiave k , restituisca il numero di nodi dell'albero che hanno chiave $< k$.
- b) Si indichino, motivando la risposta, le complessità dell'algoritmo proposto al caso pessimo e al caso ottimo.
- c) Si scriva lo pseudocodice di un algoritmo che risolve lo stesso problema di cui al punto a) sapendo che l'albero è un ABR.
- d) Si dica, motivando la risposta, se al caso ottimo la complessità dell'algoritmo proposto al punto c) migliora quella del punto b).
- e) Si dica, motivando la risposta, se al caso pessimo la complessità dell'algoritmo proposto al punto c) migliora quella del punto b).

Programmazione e Algoritmica - B (2)
Terza prova in itinere, 20 Marzo 2024

Cognome Nome:

N. Matricola:

Esercizio 1 – Tabelle Hash

Si mostri la Tabella hash ottenuta inserendo le chiavi 3, 12, 16, 14, 111,1 utilizzando un doppio hash $h(k,i) = (h_1(k) + i \cdot h_2(k)) \bmod m$ in una tabella di dimensione $m = 11$, dove $h_1(k) = k \bmod m$, e $h_2(k) = 1 + (k \bmod (m - 1))$, indicando per ogni chiave k:

- a) $h_1(k)$ ogni volta che viene calcolato,
- b) $h_2(k)$ ogni volta che viene calcolato,
- c) la sequenza di ispezione
- d) i conflitti eventualmente generati.

Esercizio 2 – Semantica statica e dinamica delle funzioni

Dato il seguente frammento di codice L,

```
var a: Int = 4;
function quiz(const x: Bool, var x: Int) -> Int {
  var res: Int = x;
  if (x == false) res = res - a;
  return res;
}
var a: Int = 10;
print(quiz(false, 18));
```

- 6. Si **dica** se il frammento di codice è corretto staticamente o no. Se non è corretto, si **dica** perché e si **corregga** l'errore
- 7. Applicando le regole (FS3) di semantica statica delle funzioni, si **costruisca l'ambiente statico** Δ_0 **associato ai parametri formali** della funzione `quiz`
- 8. Si **indichi** il contenuto dell'**ambiente statico** dopo la dichiarazione della funzione
- 9. In relazione **all'analisi dinamica**, si **indichi** il contenuto **dell'ambiente dinamico e della memoria** dopo la dichiarazione della funzione `quiz` nel caso di **scoping statico** e nel caso di **scoping dinamico** (**indicando** chiaramente la lambda- astrazione nei due casi)
- 10. Si **dica** infine se ci sono differenze fra lo scoping statico e lo scoping dinamico, **giustificando** la risposta (senza costruire il sistema di transizioni).

Esercizio 3 – Ricorsione e iterazione

Si consideri la seguente funzione ricorsiva, in cui u rappresenta il nodo di un **albero binario di ricerca**

```
Mistero (u) {  
    if (u == NIL) return NIL;  
    if (u.left != NIL) return Mistero(u.left);  
    else return u;  
}
```

1. Si **dica** quale problema risolve la funzione
2. Si **dica** se la funzione proposta è **ricorsiva in coda** o meno, **motivando** la risposta.
3. **Si scriva** una funzione iterativa che risolve lo stesso problema.

Esercizio 4 – Alberi Binari

- a) Si scriva lo pseudocodice di un algoritmo che, dati un albero binario (ovvero data la sua radice u) e una chiave k , restituisca il numero di nodi dell'albero che hanno chiave maggiore o uguale a k .
- b) Si indichino, motivando la risposta, le complessità dell'algoritmo proposto al caso pessimo e al caso ottimo.
- c) Si scriva lo pseudocodice di un algoritmo che risolve lo stesso problema di cui al punto a) sapendo che l'albero è un ABR.
- d) Si dica, motivando la risposta, se al caso ottimo la complessità dell'algoritmo proposto al punto c) migliora quella del punto b).
- e) Si dica, motivando la risposta, se al caso pessimo la complessità dell'algoritmo proposto al punto c) migliora quella del punto b).