

Programmazione e Algoritmica - B (1)
Seconda prova in itinere, 11 dicembre 2023

Cognome Nome:

N. Matricola:

Esercizio 1 – Complessità

Dato il seguente algoritmo

```
MISTERO(A,n)
  if n > 1 then {
    QuickSort(A);
    for i=1 to a do MISTERO(A,n/2);
  }
```

- a) Si scriva la relazione di ricorrenza che descrive la complessità al caso pessimo quando $a = 8$;
- b) Si risolva la relazione di cui al punto precedente motivando la risposta;
- c) Si scriva la relazione di ricorrenza che descrive la complessità al caso pessimo quando $a = 2$;
- d) Si risolva la relazione di cui al punto precedente motivando la risposta;

Esercizio 2 – Strutture Dati

Si esegua la costruzione di un Max-Heap a partire dall'array $A = [6, 0, 2, 7, 8, 5, 1, 3]$, evidenziando per ogni iterazione:

- (i) il valore dell'indice i dell'iterazione;
 - (ii) l'albero che A implicitamente rappresenta mostrando ivi gli Heap che garantiscono la correttezza della chiamata di Heapify;
 - (iii) il nodo della chiamata di Heapify effettuata, e gli scambi che essa realizza;
 - (iv) l'array A al termine dell'iterazione.
- e mostrando l'albero Heap finale.

Esercizio 3 – Limiti Inferiori

Si consideri il problema che, dato un array A di n interi, chiede di restituire un qualsiasi indice i di A tale che $A[i]=3$, e di restituire -1 se tale indice non esiste.

- a) Si dica, motivando la risposta, se il criterio della dimensione dell'input è applicabile per indicare un limite inferiore alla difficoltà del problema suddetto. Se applicabile lo si riporti e si dica se si tratta di un limite inferiore significativo.
- b) Si indichi, motivando la risposta, il limite inferiore che si ottiene applicando il criterio dell'albero di decisione. Si dica se si tratta di un limite inferiore significativo.
- c) Si dica, motivando la risposta, se il criterio della dimensione dell'input è applicabile per indicare un limite inferiore alla difficoltà dello stesso problema assumendo che A sia ordinato. Se applicabile lo si riporti e si dica se si tratta di un limite inferiore significativo.

Esercizio 4 - Semantica dinamica

Sia dato il seguente frammento di codice L:

```
const x: Int = 4;
var res: Int = -x;
if (res < x) {res = x - res;}
else {res = res - x;}
```

1. **Costruire il sistema di transizioni di stato** relativo al frammento di codice (partendo da ambiente dinamico e memoria vuoti), **indicando** chiaramente il contenuto dell'ambiente dinamico e della memoria, ad ogni passo
2. **Dimostrare** formalmente, utilizzando le regole della semantica dinamica di L, la transizione relativa all'esecuzione del comando `if`

Esercizio 5 - Funzioni: identificatori liberi, scoping statico e dinamico

Si consideri il seguente programma L:

```
var b: Int = 2;
function test (var a: Int) -> Int {
  var res: Int = a;
  res = a + b;
  return res;
};
var b: Int = 5;
print(test(b));
```

1. **Elencare, calcolandoli** con le regole FI, gli identificatori liberi della funzione, se presenti.
2. **Mostrare le transizioni di stato** dell'intero frammento di codice (partendo da ambiente dinamico e memoria vuoti) in caso di scoping statico, **indicando** chiaramente il contenuto dell'ambiente dinamico e della memoria, ad ogni passo, e il valore stampato.
3. **Determinare** il valore stampato in caso di scoping dinamico, senza costruire il sistema di transizioni, ma **motivando** la risposta.

Programmazione e Algoritmica - B (2)
Seconda prova in itinere, 11 dicembre 2023

Cognome Nome:

N. Matricola:

Esercizio 1 – Complessità

Dato il seguente algoritmo:

```
PIPP0(A,n)
if n > 1 then {
    InsertionSort(A);
    for i=1 to a do PIPP0(A,n/3);
}
```

- a) Si scriva la relazione di ricorrenza che descrive la complessità al caso pessimo quando $a=27$;
- b) Si risolva la relazione di cui al punto precedente motivando la risposta;
- c) Si scriva la relazione di ricorrenza che descrive la complessità al caso pessimo quando $a=3$;
- d) Si risolva la relazione di cui al punto precedente motivando la risposta;

Esercizio 2 – Strutture Dati

Si esegua la costruzione di un Max-Heap a partire dall'array $A=[8,0,3,10,11,7,2,5]$, evidenziando per ogni iterazione:

- (i) il valore dell'indice i dell'iterazione;
- (ii) l'albero che A implicitamente rappresenta mostrando ivi gli Heap che garantiscono la correttezza della chiamata di Heapify;
- (iii) il nodo della chiamata di Heapify effettuata, e gli scambi che essa realizza;
- (iv) l'array A al termine dell'iterazione.

e mostrando l'albero Heap finale.

Esercizio 3 – Limiti Inferiori

Si consideri il problema che, dati un array A di n interi e un intero k , chiede di restituire un qualsiasi indice i di A tale che $A[i] > k$, e di restituire -1 se tale indice non esiste.

- a) Si dica, motivando la risposta, se il criterio della dimensione dell'input è applicabile per indicare un limite inferiore alla difficoltà del problema suddetto. Se applicabile lo si riporti e si dica se si tratta di un limite inferiore significativo.
- b) Si indichi, motivando la risposta, il limite inferiore che si ottiene applicando il criterio dell'albero di decisione. Si dica se si tratta di un limite inferiore significativo.
- c) Si dica, motivando la risposta, se il criterio della dimensione dell'input è applicabile per indicare un limite inferiore alla difficoltà dello stesso problema assumendo che A sia ordinato. Se applicabile lo si riporti e si dica se si tratta di un limite inferiore significativo.

Esercizio 4 - Semantica dinamica

Sia dato il seguente frammento di codice L:

```
var ctr: Int = 3;
const x: Int = ctr + 2;
while (ctr > x) {ctr = ctr - x;}
ctr = ctr * x;
```

1. **Costruire il sistema di transizioni di stato** relativo al frammento di codice (partendo da ambiente dinamico e memoria vuoti), **indicando** chiaramente il contenuto dell'ambiente dinamico e della memoria, ad ogni passo
2. **Dimostrare** formalmente, utilizzando le regole della semantica dinamica di L, la transizione relativa all'esecuzione del comando `while`

Esercizio 5 - Funzioni: identificatori liberi, scoping statico e dinamico

Si consideri il seguente programma L:

```
var a: Int = 12;
function test (const x: Int) -> Int {
  var res: Int = x;
  res = a + x;
  return res;
};
var a: Int = 8;
print(test(a));
```

1. **Elencare, calcolandoli** con le regole FI, gli identificatori liberi della funzione, se presenti.
2. **Mostrare le transizioni di stato** dell'intero frammento di codice (partendo da ambiente dinamico e memoria vuoti) in caso di scoping statico, **indicando** chiaramente il contenuto dell'ambiente dinamico e della memoria, ad ogni passo, e il valore stampato.
3. **Determinare** il valore stampato in caso di scoping dinamico, senza costruire il sistema di transizioni, ma **motivando** la risposta.

Programmazione e Algoritmica - B (3)
Seconda prova in itinere, 11 dicembre 2023

Cognome Nome:

N. Matricola:

Esercizio 1 – Complessità

Dato il seguente algoritmo:

```
ALGO(A,n)
if n > 1 then {
    SelectionSort(A);
    for i=1 to a do ALGO(A,n/2);
}
```

- a) Si scriva la relazione di ricorrenza che descrive la complessità al caso pessimo quando $a=8$;
- b) Si risolva la relazione di cui al punto precedente motivando la risposta;
- c) Si scriva la relazione di ricorrenza che descrive la complessità al caso pessimo quando $a=2$;
- d) Si risolva la relazione di cui al punto precedente motivando la risposta;

Esercizio 2 – Strutture Dati

Si esegua la costruzione di un Max-Heap a partire dall'array $A = [12,0,4,14,16,10,2,6]$, evidenziando per ogni iterazione:

- (i) il valore dell'indice i dell'iterazione;
- (ii) l'albero che A implicitamente rappresenta mostrando ivi gli Heap che garantiscono la correttezza della chiamata di Heapify;
- (iii) il nodo della chiamata di Heapify effettuata, e gli scambi che essa realizza;
- (iv) l'array A al termine dell'iterazione.

e mostrando l'albero Heap finale.

Esercizio 3 – Limiti Inferiori

Si consideri il problema che, dato un array A di n interi, chiede di restituire un qualsiasi indice i di A tale che $A[i] < 0$, e di restituire -1 se tale indice non esiste.

- a) Si dica, motivando la risposta, se il criterio della dimensione dell'input è applicabile per indicare un limite inferiore alla difficoltà del problema suddetto. Se applicabile lo si riporti e si dica se si tratta di un limite inferiore significativo.
- b) Si indichi, motivando la risposta, il limite inferiore che si ottiene applicando il criterio dell'albero di decisione. Si dica se si tratta di un limite inferiore significativo.
- c) Si dica, motivando la risposta, se il criterio della dimensione dell'input è applicabile per indicare un limite inferiore alla difficoltà dello stesso problema assumendo che A sia ordinato. Se applicabile lo si riporti e si dica se si tratta di un limite inferiore significativo.

Esercizio 4 - Semantica dinamica

Sia dato il seguente frammento di codice L:

```
var x: Int = -5;
const y: Int = x + 10;
if (x > y) {x = x - y;}
else {x = x + y;}
```

1. **Costruire il sistema di transizioni di stato** relativo al frammento di codice (partendo da ambiente dinamico e memoria vuoti), **indicando** chiaramente il contenuto dell'ambiente dinamico e della memoria, ad ogni passo
2. **Dimostrare** formalmente, utilizzando le regole della semantica dinamica di L, la transizione relativa all'esecuzione del comando `if`

Esercizio 5 - Funzioni: identificatori liberi, scoping statico e dinamico

Si consideri il seguente programma L:

```
var x: Int = 4;
function test (var y: Int) -> Int {
  var res: Int = y;
  res = x + y;
  return res;
};
var x: Int = 2;
print(test(x));
```

1. **Elencare, calcolandoli** con le regole FI, gli identificatori liberi della funzione, se presenti.
2. **Mostrare le transizioni di stato** dell'intero frammento di codice (partendo da ambiente dinamico e memoria vuoti) in caso di scoping statico, **indicando** chiaramente il contenuto dell'ambiente dinamico e della memoria, ad ogni passo, e il valore stampato.
3. **Determinare** il valore stampato in caso di scoping dinamico, senza costruire il sistema di transizioni, ma **motivando** la risposta.