

Programmazione e Algoritmica - B (1)

Prima prova in itinere, 6 novembre 2023

Cognome Nome:

N. Matricola:

Esercizio 1 - Sintassi

Usare le produzioni della grammatica

```
E ::= E + T | T  
T ::= T * F | F  
F ::= a | ... | z | (E)
```

per costruire una **derivazione canonica destra** della stringa

$(a + b) * c + d$

Costruire inoltre **l'albero di sintassi astratta**.

Esercizio 2 - Semantica statica

Si consideri il seguente frammento di codice

```
const a: Int = 3;  
var b: Bool = false;  
while (!b) {a = a + 3;}
```

Partendo dall'ambiente statico vuoto, **si confuti o si dimostri formalmente** la seguente affermazione: *il frammento riportato è corretto staticamente*

Esercizio 3 – Problem Solving con Algoritmo Iterativo

1. Scrivere lo pseudocodice di un algoritmo iterativo efficiente che, dato un array A di n bit (ovvero di elementi dell'insieme {0,1}), lo ordini *in loco*¹.
2. Si indichi la complessità in tempo, sia al caso ottimo che al caso pessimo, dell'algoritmo proposto, motivando la risposta.

Esercizio 4 – Problem Solving con Divide et Impera

1. Scrivere lo pseudocodice di un algoritmo divide et impera efficiente che, dato un array A di n interi, restituisca la somma di tutti i valori di A, dividendolo in modo bilanciato.
2. Si scriva la relazione di ricorrenza che descrive la complessità T(n) dell'algoritmo proposto.
3. Si mostri l'albero di ricorsione che rappresenta la suddetta ricorrenza.
4. Si utilizzi l'albero di ricorsione di cui al punto 3 (oppure il metodo iterativo) per risolvere la relazione di ricorrenza di cui al punto 2, ovvero per indicare T(n) in forma esplicita.

¹ Un algoritmo ordina in loco se lo spazio utilizzato oltre all'input ha dimensione costante.

Programmazione e Algoritmica - B (2)

Prima prova in itinere, 6 novembre 2023

Cognome Nome:

N. Matricola:

Esercizio 1 – Problem Solving con Algoritmo Iterativo

1. Scrivere lo pseudocodice di un algoritmo iterativo efficiente che, dato un array A di n elementi dell'insieme {a,b}, lo ordini lessicograficamente *in loco*².
2. Si indichi la complessità in tempo, sia al caso ottimo che al caso pessimo, dell'algoritmo proposto, motivando la risposta.

Esercizio 2 – Problem Solving con Divide et Impera

1. Scrivere lo pseudocodice di un algoritmo divide et impera efficiente che, dato un array A di n interi, restituisca il valore minimo di A, dividendolo in modo bilanciato.
2. Si scriva la relazione di ricorrenza che descrive la complessità T(n) dell'algoritmo proposto.
3. Si mostri l'albero di ricorsione che rappresenta la suddetta ricorrenza.
4. Si utilizzi l'albero di ricorsione di cui al punto 3 (oppure il metodo iterativo) per risolvere la relazione di ricorrenza di cui al punto 2, ovvero per indicare T(n) in forma esplicita.

Esercizio 3 - Sintassi

Usare le produzioni della grammatica

```
E ::= E + T | T  
T ::= T * F | F  
F ::= a | ... | z | (E)
```

per costruire una **derivazione canonica destra** della stringa

$a * e + (a + b)$

Costruire inoltre **l'albero di sintassi astratta**.

Esercizio 4 - Semantica statica

Si consideri il seguente frammento di codice

```
const x: Int = 2;  
var y: Bool = false;  
if (!y) {x = x * 10;}
```

Partendo dall'ambiente statico vuoto, **si confuti o si dimostri formalmente** la seguente affermazione: *il frammento riportato è corretto staticamente*

² Un algoritmo ordina in loco se lo spazio utilizzato oltre all'input ha dimensione costante.

Programmazione e Algoritmica - B (3)

Prima prova in itinere, 6 novembre 2023

Cognome Nome:

N. Matricola:

Esercizio 1 – Problem Solving con Algoritmo Iterativo

1. Scrivere lo pseudocodice di un algoritmo iterativo efficiente che, dato un array A di n elementi dell'insieme {-1,1}, modifichi l'ordine degli elementi di A mettendo tutti i '-1' addossati a sinistra e gli '1' a destra, operando *in loco*³.
2. Si indichi la complessità in tempo, sia al caso ottimo che al caso pessimo, dell'algoritmo proposto, motivando la risposta.

Esercizio 2 - Sintassi

Usare le produzioni della grammatica

```
E ::= E + T | T  
T ::= T * F | F  
F ::= a | ... | z | (E)
```

per costruire una **derivazione canonica destra** della stringa

a * (c + d) + b

Costruire inoltre l'**albero di sintassi astratta**.

Esercizio 3 – Problem Solving con Divide et Impera

1. Scrivere lo pseudocodice di un algoritmo divide et impera efficiente che, dato un array A di n interi, restituisca T se la somma di tutti i valori di A è pari, e restituisca F altrimenti, dividendolo in modo bilanciato.
2. Si scriva la relazione di ricorrenza che descrive la complessità T(n) dell'algoritmo proposto.
3. Si mostri l'albero di ricorsione che rappresenta la suddetta ricorrenza.
4. Si utilizzi l'albero di ricorsione di cui al punto 3 (oppure il metodo iterativo) per risolvere la relazione di ricorrenza di cui al punto 2, ovvero per indicare T(n) in forma esplicita.

Esercizio 4 - Semantica statica

Si consideri il seguente frammento di codice

```
var a: Bool = false;  
const b: Bool = true;  
if (!a) {b = a || b;}
```

Partendo dall'ambiente statico vuoto, **si confuti o si dimostri formalmente** la seguente affermazione: *il frammento riportato è corretto staticamente*

³ Un algoritmo ordina in loco se lo spazio utilizzato oltre all'input ha dimensione costante.

Programmazione e Algoritmica - B (4)

Prima prova in itinere, 6 novembre 2023

Cognome Nome:

N. Matricola:

Esercizio 1 - Semantica statica

Si consideri il seguente frammento di codice

```
const x: Double = 3.2;
var y: Bool = false;
while (!y) {x = x * 1.2;}
```

Partendo dall'ambiente statico vuoto, **si confuti o si dimostri formalmente** la seguente affermazione: *il frammento riportato è corretto staticamente*

Esercizio 2 – Problem Solving con Divide et Impera

1. Scrivere lo pseudocodice di un algoritmo divide et impera efficiente che, dato un array A di n interi, restituisca il valore del numero pari più alto, dividendolo in modo bilanciato.
2. Si scriva la relazione di ricorrenza che descrive la complessità T(n) dell'algoritmo proposto.
3. Si mostri l'albero di ricorsione che rappresenta la suddetta ricorrenza.
4. Si utilizzi l'albero di ricorsione di cui al punto 3 (oppure il metodo iterativo) per risolvere la relazione di ricorrenza di cui al punto 2, ovvero per indicare T(n) in forma esplicita.

Esercizio 3 - Sintassi

Usare le produzioni della grammatica

```
E ::= E + T | T
T ::= T * F | F
F ::= a | ... | z | (E)
```

per costruire una **derivazione canonica destra** della stringa

a + (b * c) * e

Costruire inoltre l'**albero di sintassi astratta**.

Esercizio 4 – Problem Solving con Algoritmo Iterativo

1. Scrivere lo pseudocodice di un algoritmo iterativo efficiente che, dato un array A di n booleani (ovvero di elementi nell'insieme {T,F}), modifichi l'ordine degli elementi di A mettendo tutti i T addossati a sinistra e tutti gli F a destra, operando *in loco*⁴.
2. Si indichi la complessità in tempo, sia al caso ottimo che al caso pessimo, dell'algoritmo proposto, motivando la risposta.

⁴ Un algoritmo ordina in loco se lo spazio utilizzato oltre all'input ha dimensione costante.