

# Progetto ASD

Ricostruzione di alberi dalle visite

Algoritmi e Strutture Dati

Anno Accademico 2025/2026

# IL PROBLEMA: RICOSTRUZIONE DI ALBERI

## OBIETTIVO

Dato un **albero binario** rappresentato tramite due visite, ricostruire la struttura completa dell'albero.

## VISITE DISPONIBILI (INPUT GARANTITO)

Vengono fornite sempre due visite. Una è **necessariamente In-order**. L'altra può essere Pre-order o Post-order.

- **Pre-order + In-order**
- **Post-order + In-order**

## STRUTTURA DEL FILE INPUT.TXT

$N$

$VT_1 \ n_{11} \ n_{12} \ n_{13} \ \dots \ n_{1N}$

$VT_2 \ n_{21} \ n_{22} \ n_{23} \ \dots \ n_{2N}$

- $N$ : numero di nodi dell'albero
- $VT_1, VT_2$ : tipo di visita
  - ▶  $-1$  = Pre-order
  - ▶  $0$  = In-order (Sempre presente)
  - ▶  $1$  = Post-order
- $n_{x1} \dots n_{xN}$ : sequenza di nodi visitati

## ASSUNZIONI

$1 \leq N \leq 250.000$

## INPUT.TXT

10

-1 1 4 3 2 7 6 9 5 0 8

0 2 3 4 6 7 9 1 0 5 8

- 10 nodi nell'albero (ID da 0 a 9)
- Prima riga: **Preorder** (-1): 1, 4, 3, 2, 7, 6, 9, 5, 0, 8
- Seconda riga: **Inorder** (0): 2, 3, 4, 6, 7, 9, 1, 0, 5, 8

## STRUTTURA DEL FILE OUTPUT.TXT

Il file deve contenere  $N$  righe, ordinate per ID del nodo (da 0 a  $N - 1$ ). Ogni riga segue il formato:

```
ID_nodo numero_figli [figlio_sx] [figlio_dx]
```

- **ID\_nodo**: L'indice della riga (0, 1, ...,  $N-1$ )
- **numero\_figli**: 0, 1 oppure 2
- **figlio\_sx** / **dx**: Lista dei figli presenti, in ordine rigoroso (prima sinistro, poi destro)

*Nota: Se un nodo ha 0 figli, la riga termina dopo il numero di figli.*

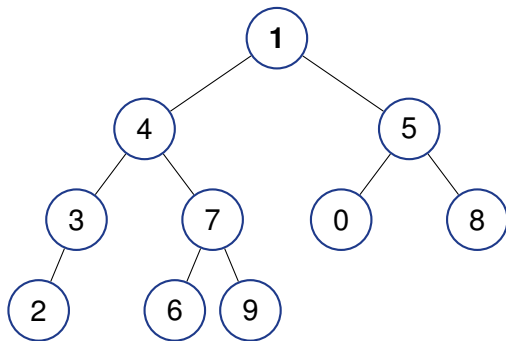
## OUTPUT.TXT

```
0 0
1 2 4 5
2 0
3 1 2
4 2 3 7
5 2 0 8
6 0
7 2 6 9
8 0
9 0
```

## Spiegazione:

- **Nodo 0:** Foglia (0 figli)
- **Nodo 1:** Radice, 2 figli (4 sx, 5 dx)
- **Nodo 3:** Ha 1 figlio (2). Dato che nell'inorder (2,3...) il 2 precede il 3, il 2 è figlio **sinistro**.

# VISUALIZZAZIONE DELL'ESEMPIO



**Pre:** 1, 4, 3, 2, 7, 6, 9, 5, 0, 8   |   **In:** 2, 3, 4, 6, 7, 9, 1, 0, 5, 8

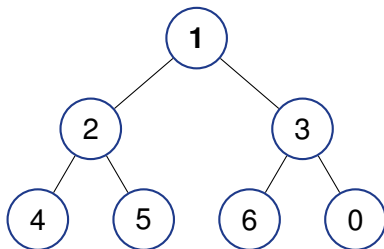
# ESEMPIO 1: PREORDER + INORDER

## INPUT

```
7
-1 1 2 4 5 3 6 0 // pre-order
0 4 2 5 1 6 3 0 // in-order
```

## OUTPUT

```
0 0
1 2 2 3
2 2 4 5
3 2 6 0
4 0
5 0
6 0
```





## ESEMPIO 2: INORDER + POSTORDER

### INPUT

9

0 8 4 2 5 0 1 6 3 7 // *in*

1 8 4 0 5 2 6 7 3 1 // *post*

### OUTPUT

0 0

1 2 2 3

2 2 4 5

3 2 6 7

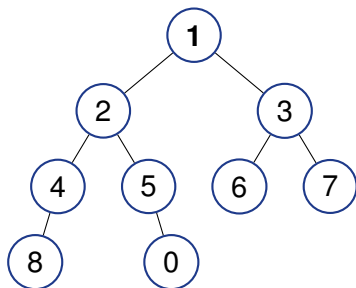
4 2 8

5 1 0

6 0

7 0

8 0



## ESEMPIO 3: ALBERO COMPLETAMENTE SBILANCIATO

### INPUT

6

-1 4 3 2 1 5 0 // *pre-order*

0 0 5 1 2 3 4 // *in-order*

### OUTPUT

0 0

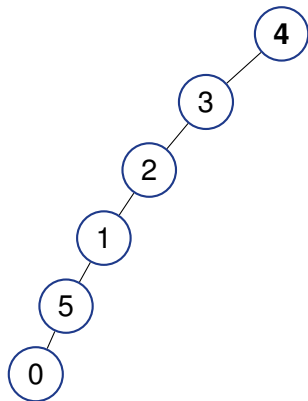
1 1 5

2 1 1

3 1 2

4 1 3

5 1 0



**Caso degenerare:** Albero completamente sbilanciato a sinistra

## SCADENZA

**martedì 16 dicembre 2025 ore 18:00**

Per caricare il vostro codice, recatevi su

<https://asdlab.disi.unitn.it/arena/>

Come username dove usare l'identificativo del vostro gruppo (gid) che trovate su questa pagina:

<https://asdlab.disi.unitn.it/asd25/gruppi/> (quindi vi loggate come `groupNNN`). La password è quella che avete specificato nel form di iscrizione.

## CLASSIFICA

La classifica è disponibile all'indirizzo:

<https://asdlab.disi.unitn.it/arena/ranking>

e tiene conto dell'**ultima** soluzione consegnata.

- Siamo disponibili per una sessione di ricevimento **martedì 16** (su richiesta)
- Rispondiamo su Telegram
- Rispondiamo alle vostre mail (scrivete direttamente agli esercitatori e ai tutor, trovate le mail sul sito)

# **Domande?**

Università degli Studi di Trento  
Dipartimento di Ingegneria e Scienze dell'Informazione

Anno Accademico 2025/2026