

# Programmazione

Docenti: Giovanni Da San Martino

Francesco Gavazzo

**Lamberto Ballan**

<[lamberto.ballan@unipd.it](mailto:lamberto.ballan@unipd.it)>

Programmazione, A.A. 2023/24

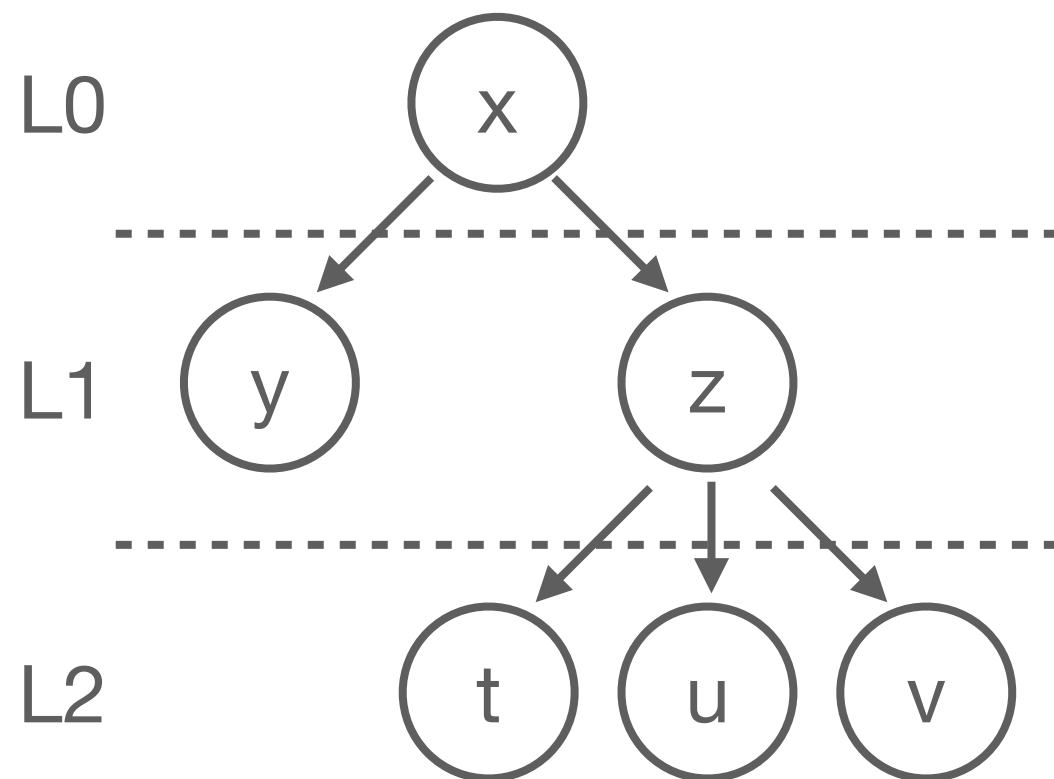
SCQ0093758 - LT Informatica



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

- Albero: è un insieme di elementi (nodi) su cui è definita una relazione di “discendenza” con due proprietà:
  - Esiste un unico nodo, detto *radice*, che non ha predecessori
  - Ogni altro nodo ha un unico predecessore
- I nodi che non hanno successori sono detti *foglie*
- L'albero è ripartito a livelli e i nodi che non sono ne la radice, ne una foglia, sono detti nodi *intermedi*

- Vediamo un esempio:



- x è il nodo radice
- z è un nodo intermedio
- y, t, u, v sono foglie
- L'albero è ripartito su tre livelli ed ha quindi profondità pari a 2
- Il nodo z realizza il massimo grado di uscita (pari a 3)

- Formalmente l'albero è definito come una coppia  $tree = \langle V, E \rangle$  dove:
  - $V$  è un insieme di valori di un qualche tipo
  - $E \subseteq V \times V$  è una relazione su  $V$ , ovvero un insieme di coppie  $e = \langle v_{parent}, v_{child} \rangle$  con *parent* e *child* elementi di  $V$ 
    - $v_{parent}$  è predecessore di  $v_{child}$  (che viceversa è successore)
  - Gli elementi di  $V$  sono detti *vertici* e quelli di  $E$  *archi*

- Nella caratterizzazione di un albero hanno rilevanza:
  - Grado di uscita di un nodo = numero dei suoi successori diretti
  - Profondità = distanza di un nodo dalla radice (che ha per definizione profondità 0)
- Per “estensione”, la profondità di un albero è data dal max delle profondità dei nodi che lo compongono
- L'insieme dei nodi a stessa profondità forma un livello
- Un albero si dice bilanciato quando su qualsiasi nodo la profondità dei sottoalberi differisce al massimo di 1

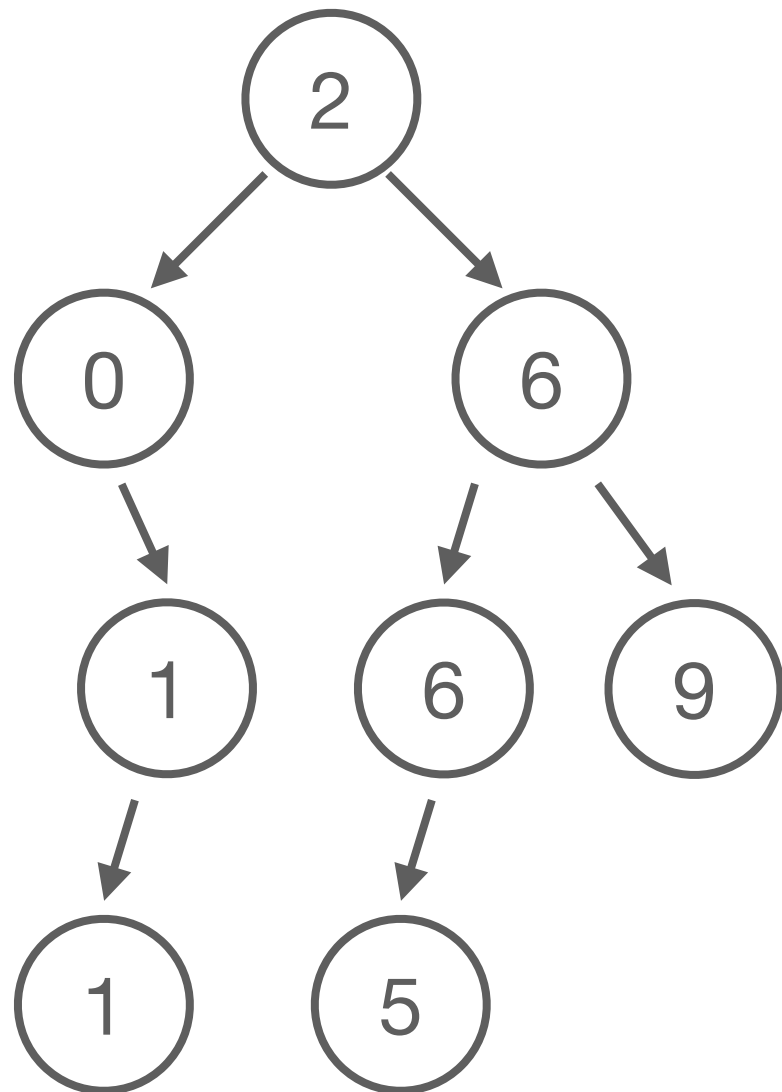
- Su un albero si eseguono le stesse operazioni che si applicano ad una lista (con un paio di differenze)...
- Su un albero non esiste un unico nodo terminale, ma si hanno invece più foglie
  - Per cui l'*inserimento in coda* deve essere qualificato con un criterio di selezione di quale foglia sia il target dell'operazione
  - L'*inserimento in testa* (o in posizione intermedia) tende a far degenerare l'albero verso una forma sequenziale; per cui gli inserimenti sono tipicamente eseguiti sulle foglie

- Un albero binario è un albero sul quale ciascun nodo ha grado di uscita minore o uguale a 2
- Un albero binario si dice di ricerca quando:
  - Il valore codificato da ciascun nodo è  $\geq$  del valore sul figlio sinistro e  $<$  stretto del valore codificato sul figlio destro

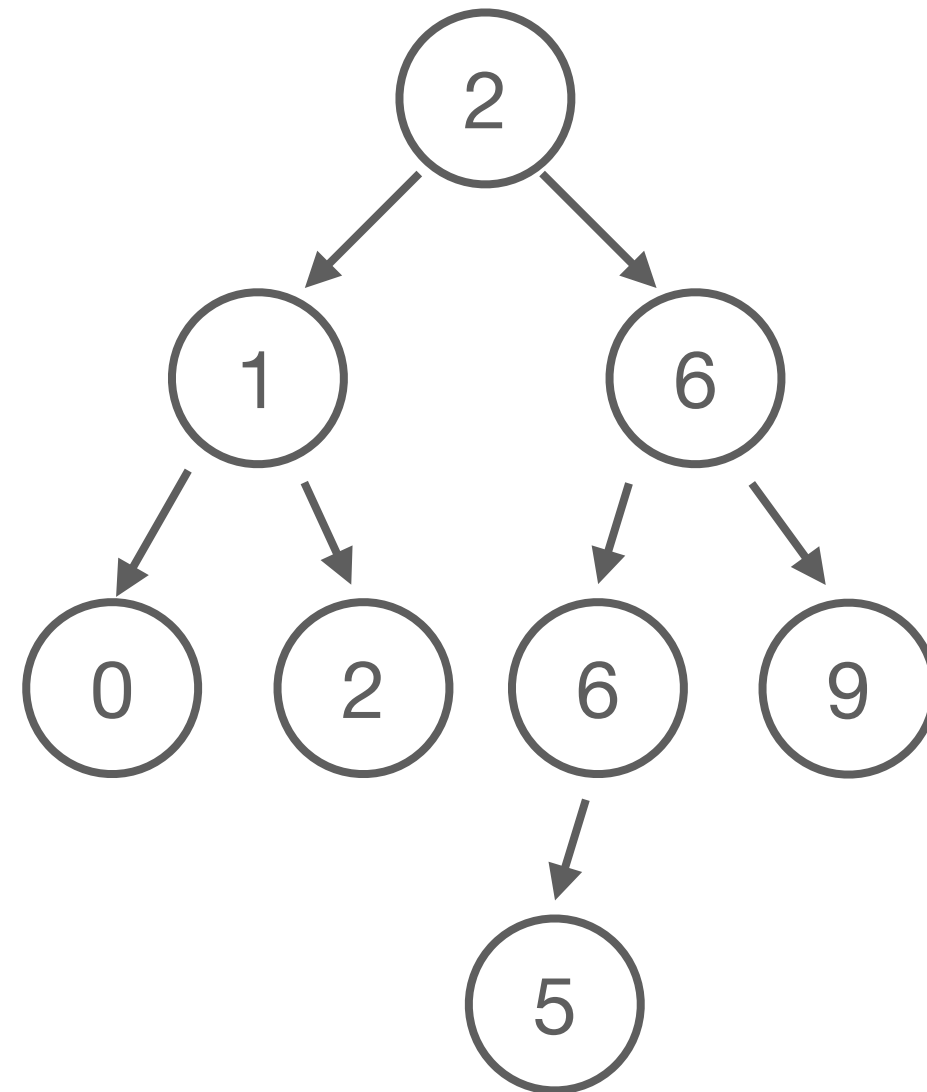
## BST ed analisi di complessità

	<u>Media</u>	<u>Worst case</u>
Spazio (in mem.):	$O(n)$	$O(n)$
Ricerca:	$O(\log n)$	$O(n)$
Inserimento:	$O(\log n)$	$O(n)$
Cancellazione:	$O(\log n)$	$O(n)$

- Vediamo due esempi:



Albero Binario di Ricerca sbilanciato

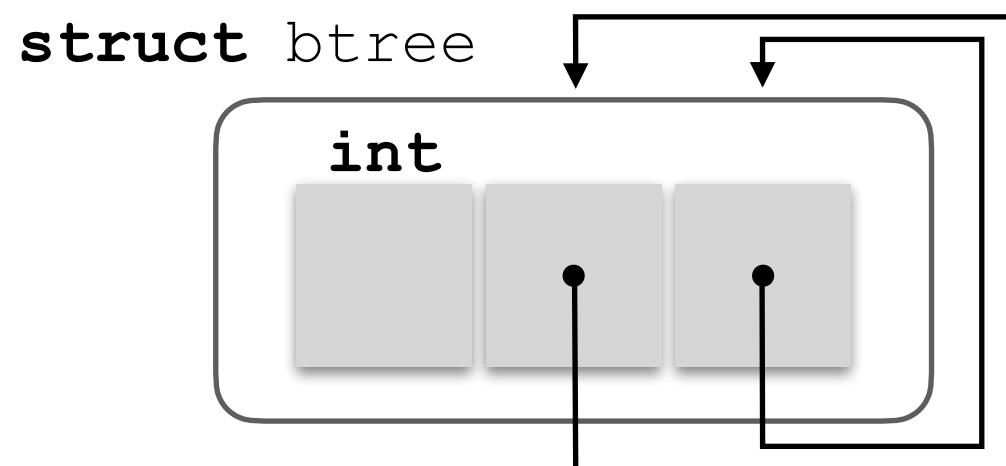


Albero Binario di Ricerca bilanciato



- Vediamo l'estensione della rappresentazione collegata con puntatori vista per le liste per un albero binario:

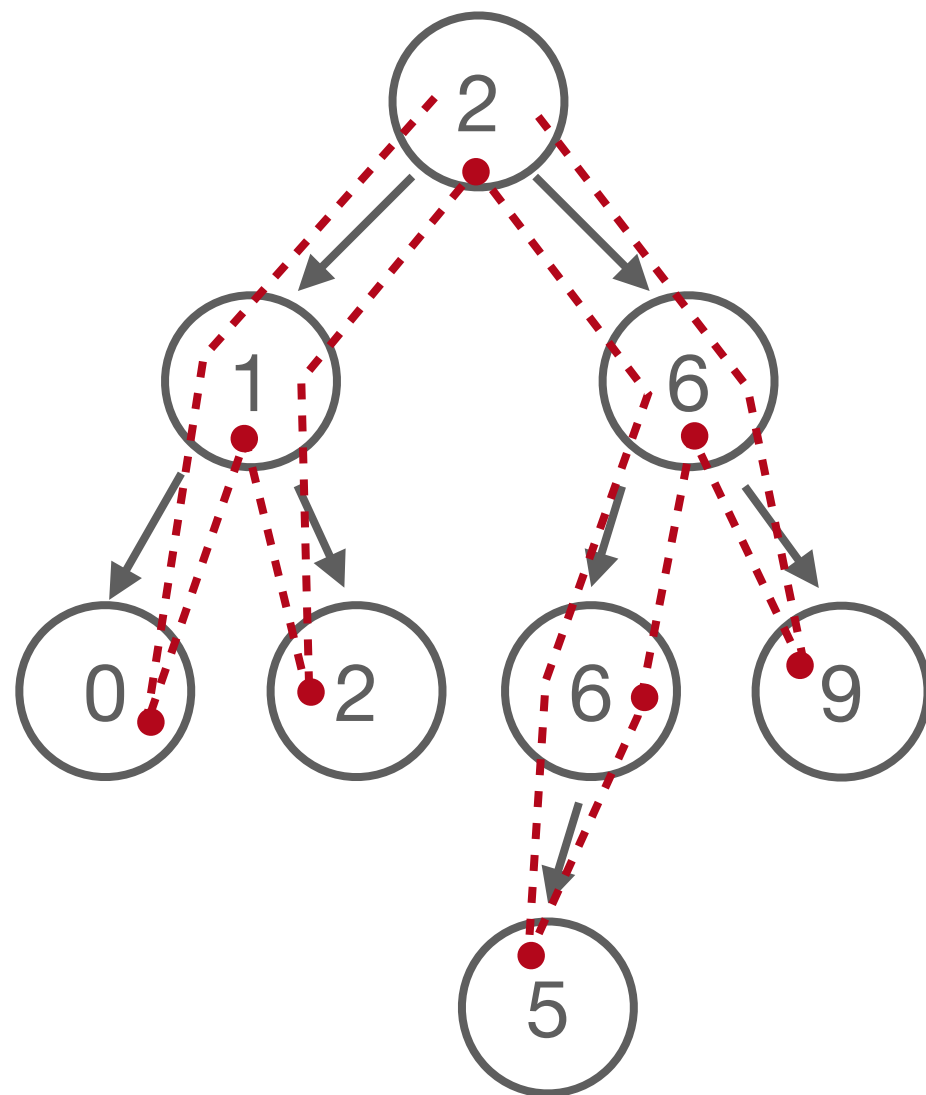
```
struct btree {  
    int value;  
    struct btree *leftPtr;  
    struct btree *rightPtr;  
};
```



- Oggi inizieremo a:
  - Sviluppare funzioni di inserimento, visita, ricerca etc. su alberi binari di ricerca
  - Vedremo in particolare implementazioni basate su ricorsione

- La visita di un albero binario di ricerca (ad esempio per funzioni di stampa) può avvenire in tre modi:
  - In forma simmetrica (ordinata)
  - In forma anticipata (pre-ordine)
  - In forma posticipata (post-ordine)
    - Nota: in questo caso il nodo “corrente” viene visitato dopo la sua discendenza (per cui viene comunemente indicata come *visita in profondità* / “depth-first”)

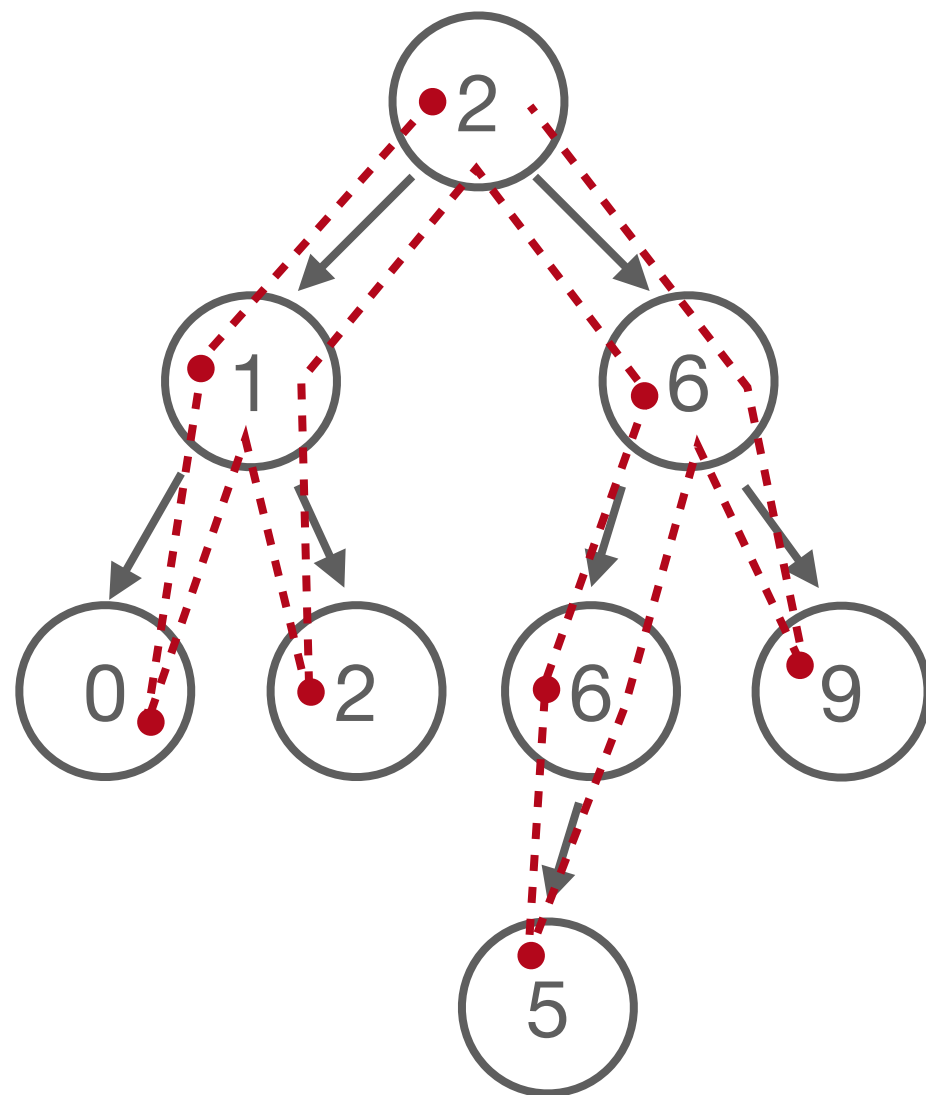
- Vediamo un esempio:



**VISITA SIMMETRICA** (*in ordine*)

Output: 0, 1, 2, 2, 5, 6, 6, 9

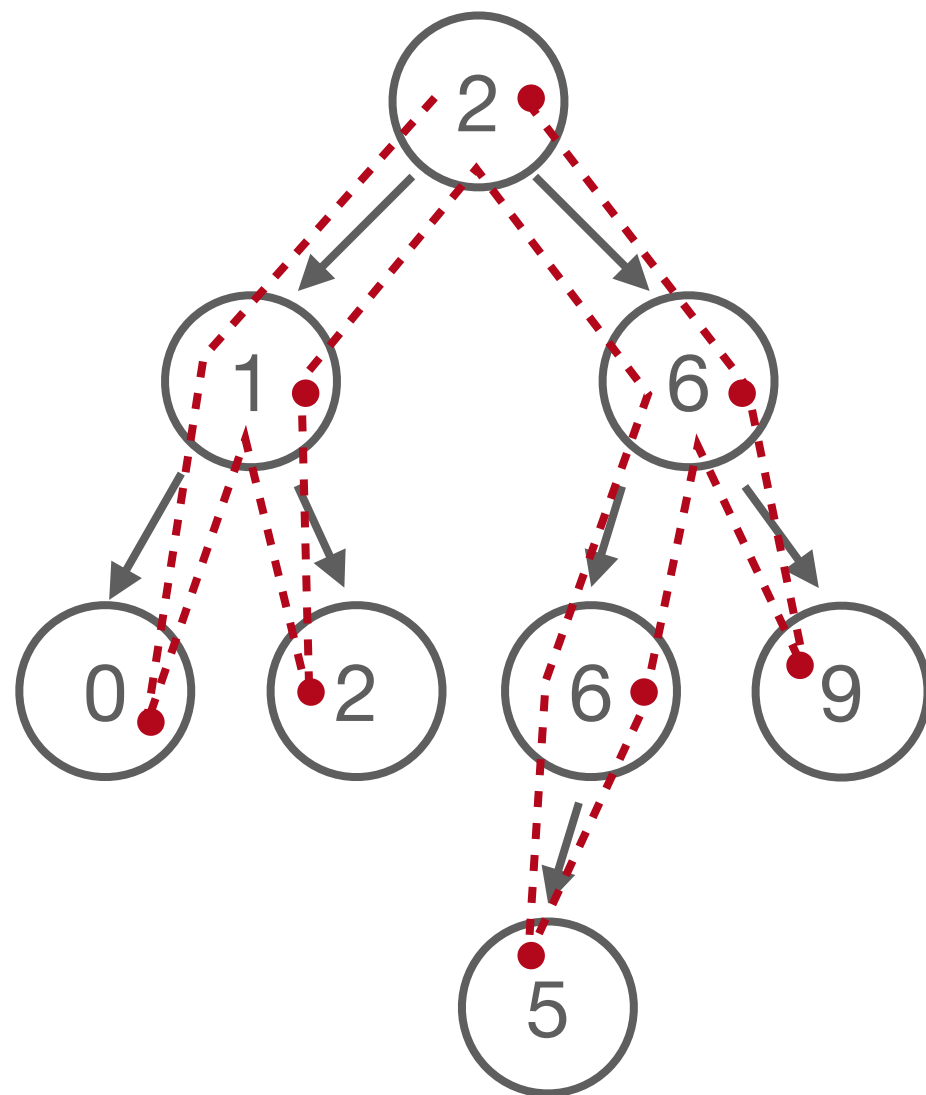
- Vediamo un esempio:



**VISITA ANTICIPATA** (*in pre-ordine*)

Output: 2, 1, 0, 2, 6, 6, 5, 9

- Vediamo un esempio:



**VISITA POSTICIPATA / IN PROFONDITA'**

Output: 0, 2, 1, 5, 6, 9, 6, 2

terminal:~\$ ■

Lavoriamo online e alla lavagna...

- **Ufficio:** Torre Archimede, ufficio 6CD3
- **Ricevimento:** ~~Venerdì 9:00-11:00~~; fino alla fine delle lezioni: Giovedì 8:30-10:30 (*inviare cmq e-mail per conferma*)

✉ [lamberto.ballan@unipd.it](mailto:lamberto.ballan@unipd.it)

🏠 <http://www.lambertoballan.net>

🏠 <http://vimp.math.unipd.it>

@ [twitter.com/lambertoballan](https://twitter.com/lambertoballan)