ESERCIZIO 1

→ $O(n+m)$

Merge based in cui scorriamo le due liste

→ $O(m \lg n)$ binary search based

In questo caso abbiamo un array con m elementi e facciamo m ricerche binarie nell'altro array lungo n

→ $O\left(m\left(1+\lg \frac{n}{m}\right)\right)$

Questo è il caso del mutual partitioning o del doubling che non usa neanche la ricorsione.

ESERCIZIO 2

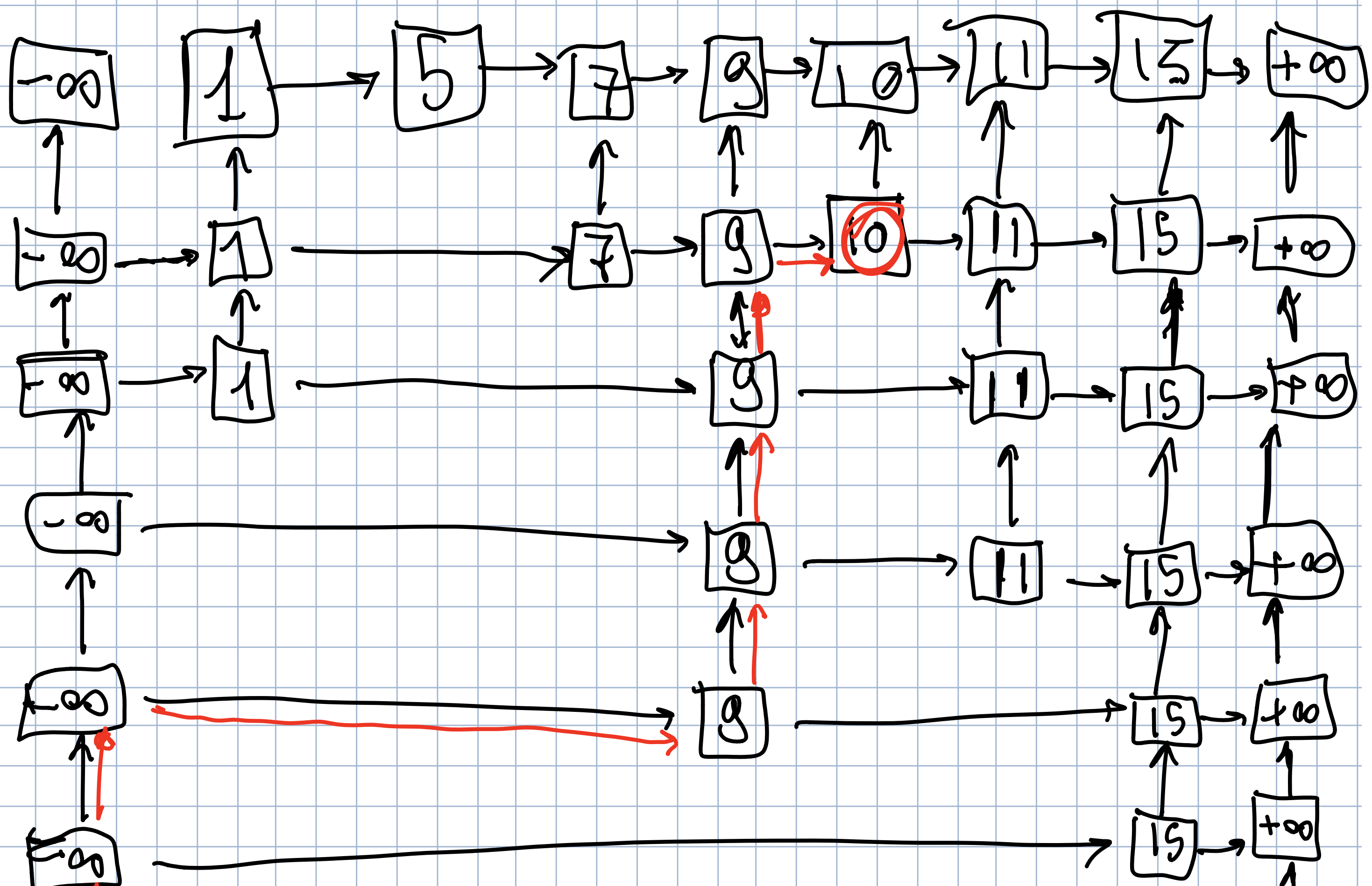
Skip list: è una struttura dati in cui abbiamo una lista di interi e vari livelli, in ogni livello troviamo la metà degli interi del livello precedente. In particolare per ogni elemento abbiamo probabilità $\frac{1}{2}$ di andare al livello successivo e $\frac{1}{2}$ di non andarci

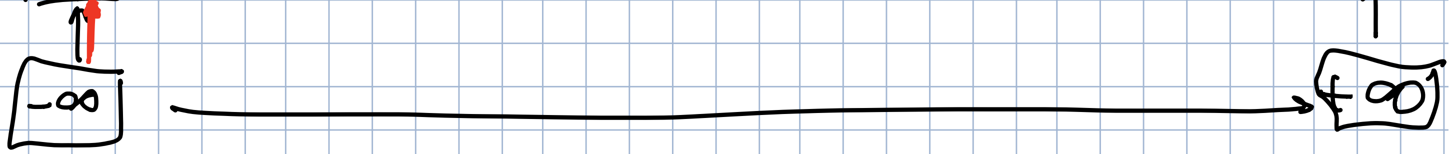
Quindi la ricerca nella skip list costa $O(\log n)$ e l'altezza della skip list è al più $\log n$.

È al più $\log n$ perché per ogni intero la prob di andare al livello successivo è $\frac{1}{2}$, $P(L \geq l) = \frac{1}{2^l}$ se abbiamo n interi diventa $\frac{n}{2^l}$ dato $l = c \log n$

allora

$$\frac{n}{2^{c \log n}} = \frac{n}{n^c} = \frac{1}{n^{c-1}} = O(\log n)$$





Ricerca di 10

ESERCIZIO 3

Per risolvere LCA in space $O(n \lg n)$ e time $O(1)$:

→ Prendiamo le due stringhe e le mettiamo una dopo l'altra e creiamo il corrispondente suffix tree. Calcoliamo l'array LCP.

Ora se vogliamo calcolare LCA sull'albero dobbiamo prendere le due foglie e calcolare il nodo che hanno in comune.

Se usiamo l'array LCP prendiamo il minimo dell'array lcp (RMQ).

Ora la struttura:

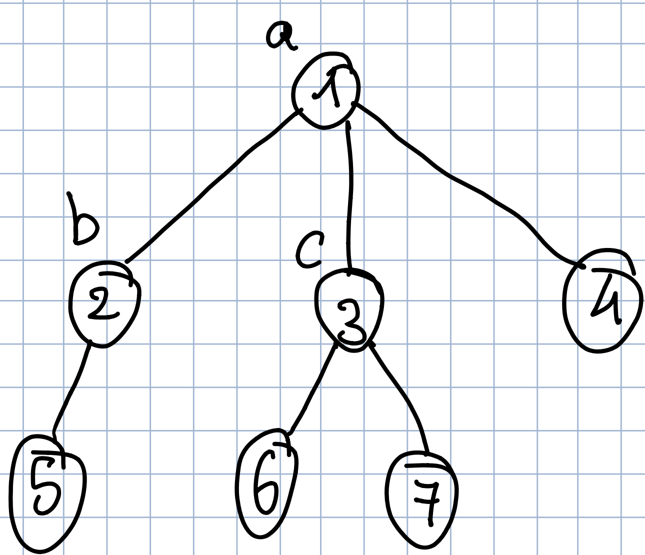
1. Svolgiamo l'Euler Tour del suffix tree ottenendo un array di dimensione $2e$ dove e = numero nodi
2. Creiamo l'array D in cui scriviamo la profondità dei nodi dell'euler tour.
3. Dividiamo l'array D in blocchi di

dimensione d. Quindi abbiamo $\frac{2\ell}{d}$ blocchi e per ogni blocco calcoliamo il minimo

④ Per ogni blocco calcoliamo il minimo e lo mettiamo in un array M.

⑤ Sull'array M applichiamo il metodo sparsify

⑥ Troviamo dove si trova il minimo



$\Delta =$ a b 5 b | a c 6 c | 7 c a 4 | a

$D =$ 1 2 3 2 | 1 2 3 2 | 3 2 1 2 | 1

Dividiamo in blocchi di dimensione $d = 4$

Per ogni blocco calcolo il minimo

M =				
	1	1	1	1