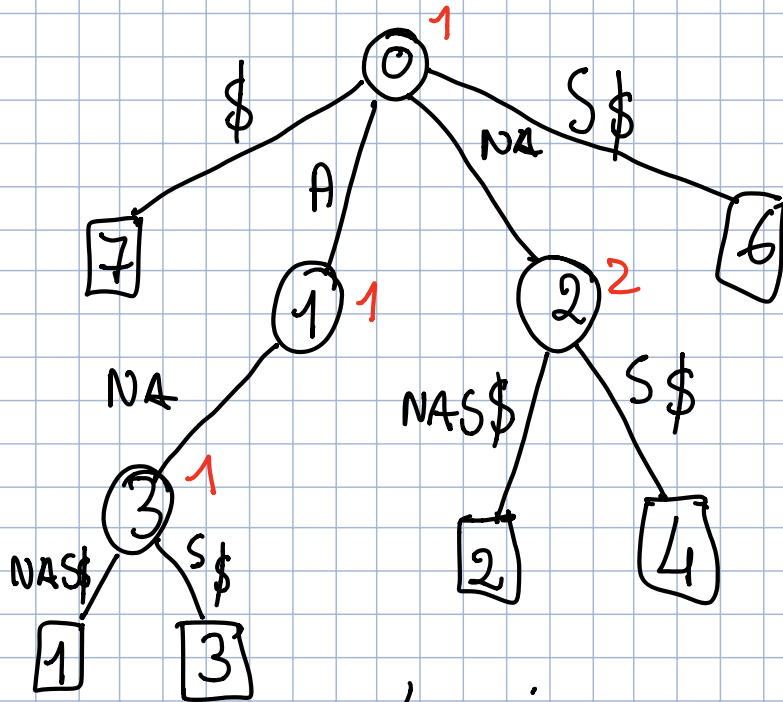


ESERCIZI 6 DICEMBRE

① $T = \text{ANANAS\$}$
using suffix tree

Compute LZ-parsing



A N A N A S \$
1 2 3 4 5 6 7

A | N | A N A S
<0,0,A> | <0,0,N> | <2,3,S>

ESERCIZIO 2

$T = \text{ABACABRA}$

$P = \text{ABR}$

Come funziona l'algoritmo 1-mismatch?

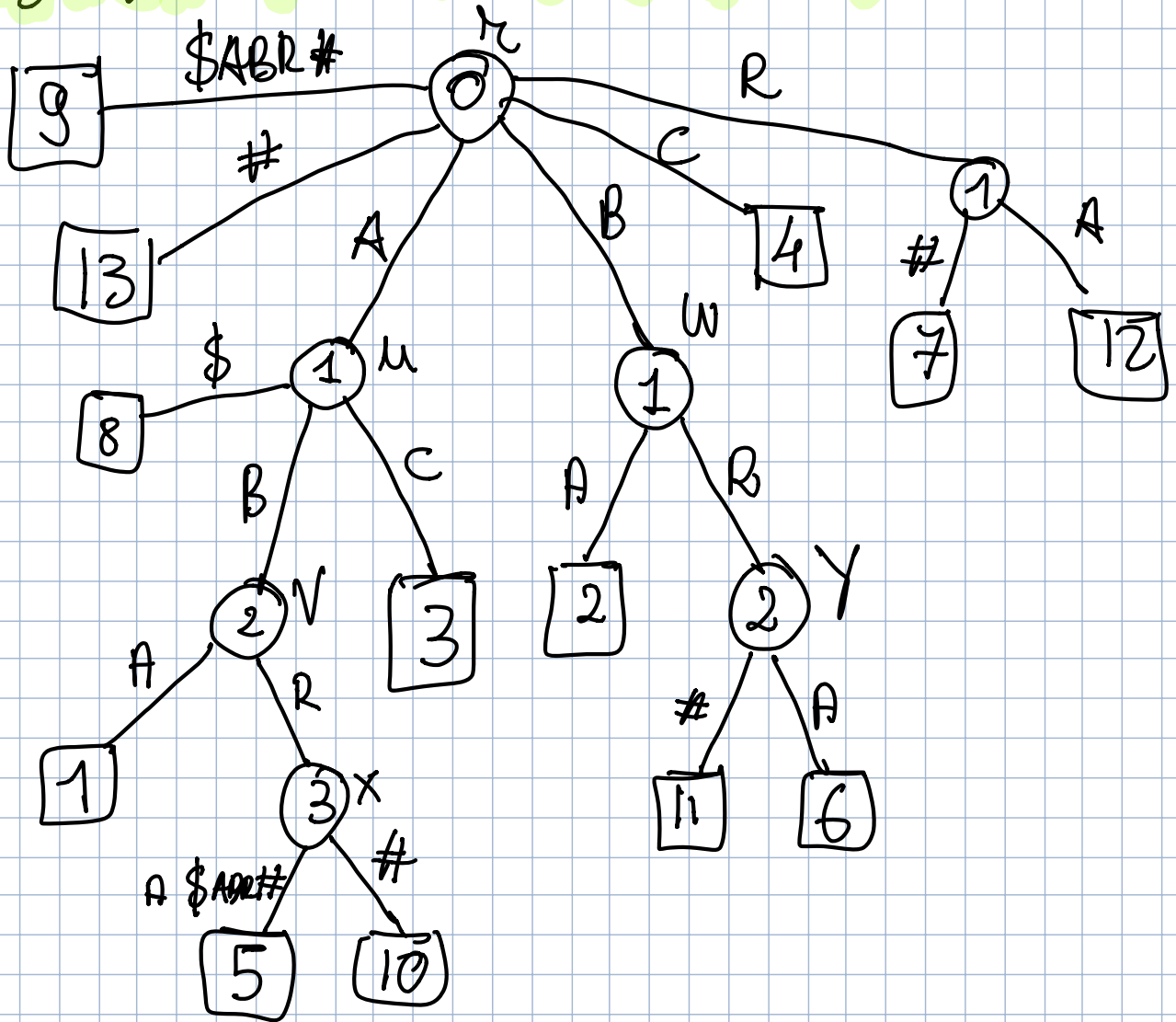
Dato una struttura che give LCA in $O(1)$ time.

② Given $T = \text{ABACABRA}$ and $P = \text{ABR}$

1) Si deve trovare $ABACABRA\$ABR$
e trovare il suffix tree.

Poi si cerca P nel suffix tree

A B A C A B R A \$ A B R #
1 2 3 4 5 6 7 8 9 10 11 12 13

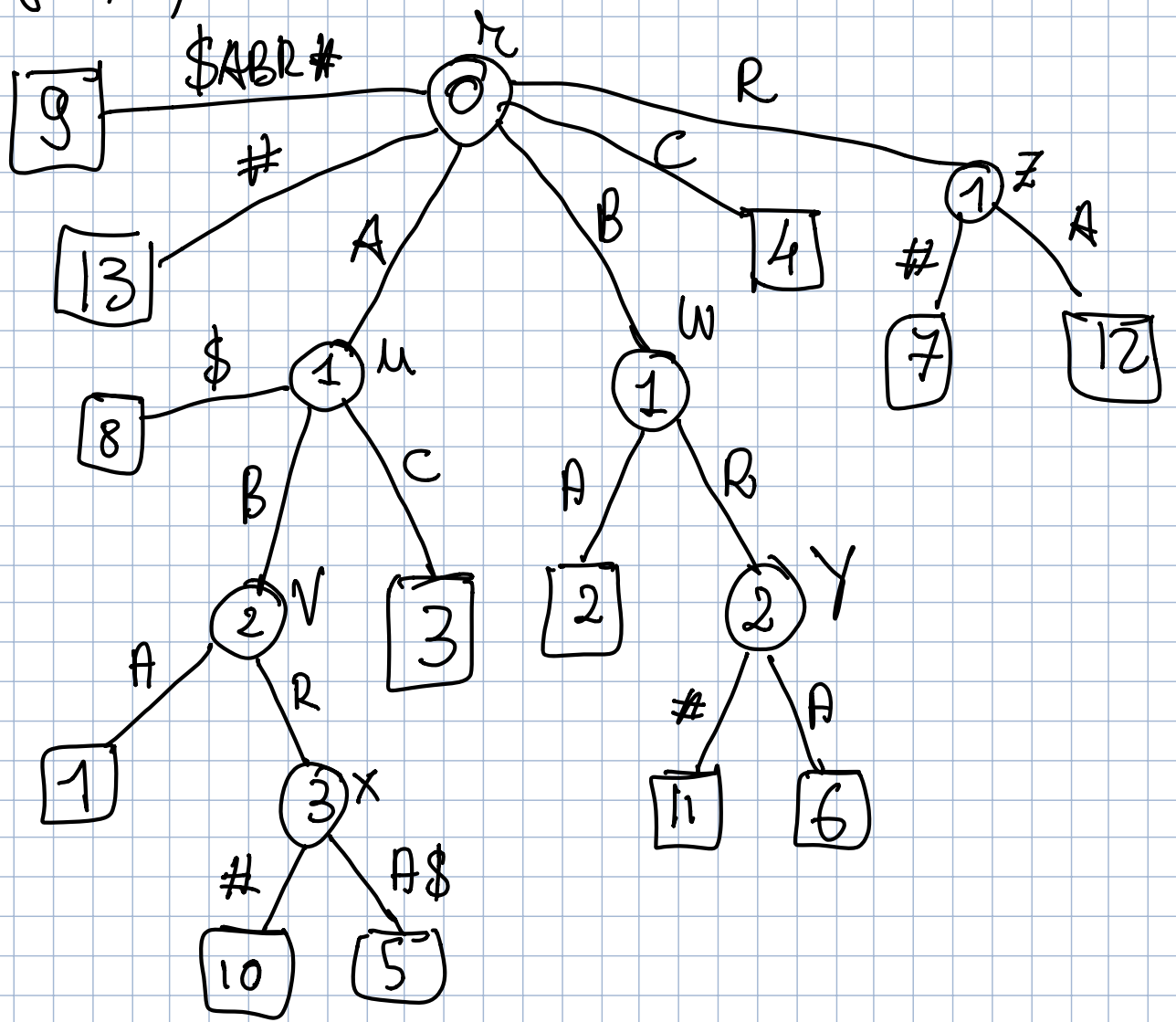


Algoritmo per calcolare il k mismatch,
viene calcolato ad ogni passaggio
$$lcp(T(i-s+1, n), P(s, p))$$

$T = A \ B \ A \ C \ A \ B \ R \ A \ \$ \ A \ B \ R \ #$
 $P = A \ B \ R$

- $\bullet \text{ lca}(T[1, 9], P[10, 13]) = 2$, mismatch su
 $A \neq R$ poi mi fermo
 $\text{matches} = \{T[1, 3]\}$
- $\bullet \text{ lca}(T[2, 9], P[10, 13]) =$ mismatch su
 $B \neq A, A \neq B$ mi fermo
- $\bullet \text{ lca}(T[3, 9], P[10, 13]) = 1$ mismatch su $B \neq C$
 $\text{e } R \neq A$
- $\bullet \text{ lca}(T[4, 9], P[10, 13]) =$ mismatch $C \neq A$
 $A \neq B$
- $\bullet \text{ lca}(T[5, 9], P[10, 13]) =$ match perfetto
 $\text{matches} = \{T[1, 3], T[5, 9]\}$

Provide a data structure that allows
 LCA in constant time and compute
 LCA(10.6) block 4. Non Consideriamo \$ e #



E.T. r u 8 u v 1 v x 10 x 5 x v u 3 u r w 2 w
 y 11 y 6 y w r 4 r z 7 z 12 z r

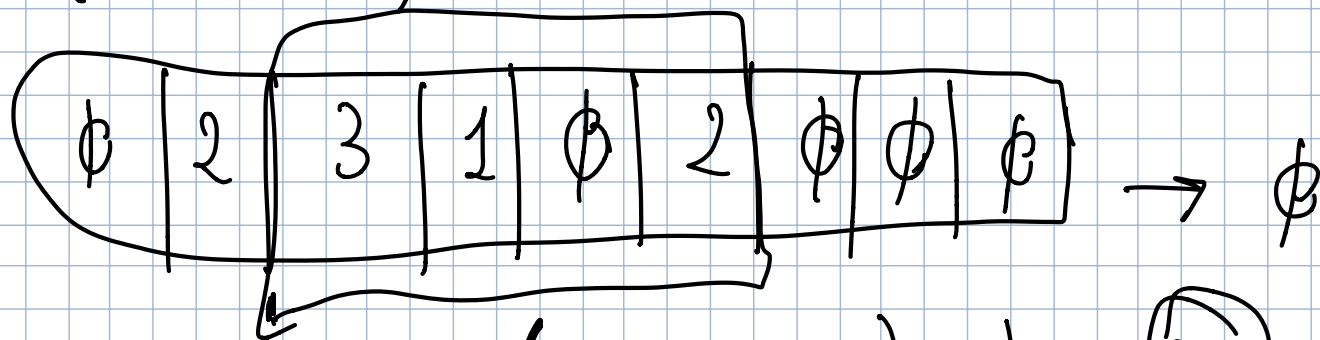
0 | 1 | 2 | 1 | 2 | 3 | 2 | 3 | 4 | 3 | 2 | 1 | 2 | 1
 10 | 1 | 2 | 1 | 2 | 3 | 2 | 3 | 2 | 1 | 0 | 1 | 0 | 1
 2 | 1 | 2 | 1 | 0

3	1	0	2
---	---	---	---

lca(10, 6) → devo prendere il 10 più a sinistra e il 6 più a destra. quindi consideriamo il RHE tra i blocchi 3, 4, 5, 6

Prendiamo come 2^L la più grande potenza di 2 < lunghezza.

$$(i, i + 2^L - 1) = (3, 3 + 2 - 1) = 1$$



$$(6 - 2 + 1, 6) = \emptyset = \pi$$