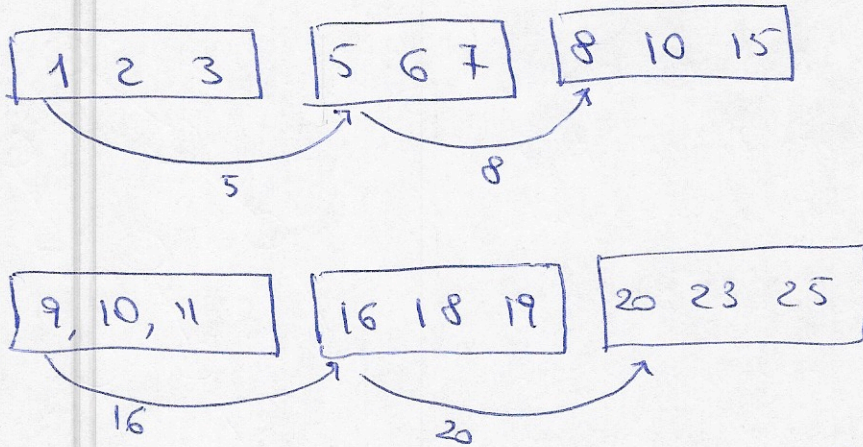


# Information Retrieval

15/1/21

Q1.



Recall that first we compare the items of the lists and then, in case of difference, then check the skip pointers in the ~~previous~~ <sup>current</sup> first item of a block. We indicate with  $\boxed{x}$  the skip pointer

$\langle 1, 9 \rangle$     $\langle \boxed{5}, 9 \rangle$     $\langle \boxed{8}, 9 \rangle$     $\langle \boxed{+\infty}, 9 \rangle$     $\langle 10, 9 \rangle$     $\langle 10, \boxed{16} \rangle$   
 $\langle 10, 10 \rangle$

Q2.

	2x mod 13	3y mod 13
A = 1, 4, 6, 8	②, 8, 12, 3	③, 12, 5, 11
B = 2, 4, 6, 7, 8	4, 8, 12, ①, 3	6, 12, ⑤, 8, 11
C = 4, 6, 11	⑧, 12, 9	⑫, ⑤, 7

$$\left. \begin{array}{l}
 \text{sketch (A)} = [2, 3] \\
 \text{sketch (B)} = [1, 5] \\
 \text{sketch (C)} = [8, 5]
 \end{array} \right\} \begin{array}{l}
 J(A, B) = 0/2 = 0 \\
 J(A, C) = 0/2 = 0 \\
 J(B, C) = 1/2
 \end{array}$$

Q3

$$D = \{BAS, BOX, BUS, CUS\}$$

$$D_1 = \{ BA [BAS], BS [BAS], AS [BAS] \\ BO [BOX], BX [BOX], OX [BOX] \\ BU [BUS], BS [BUS], US [BUS] \\ CU [CUS], CS [CUS], US [CUS] \}$$

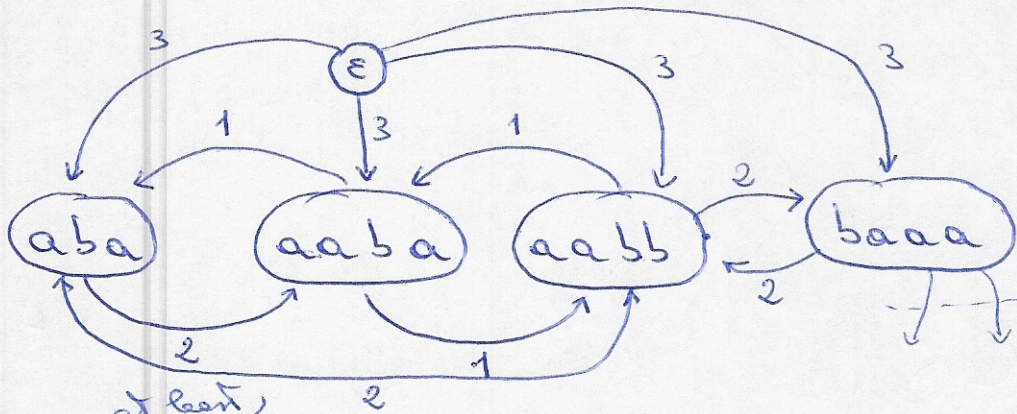
$P = ROX \rightarrow$  Search  $P$  in  $D \rightarrow \emptyset$

Search  $P^-$  in  $D \rightarrow$  search  $RO, RX, OX$  in  $D \rightarrow \emptyset$

Search  $P$  in  $D_1 \rightarrow \emptyset$

Search  $P^-$  in  $D_1 \rightarrow$   
search  $RO \rightarrow \emptyset$   
search  $RX \rightarrow \emptyset$   
search  $OX \rightarrow BOX$

Q4

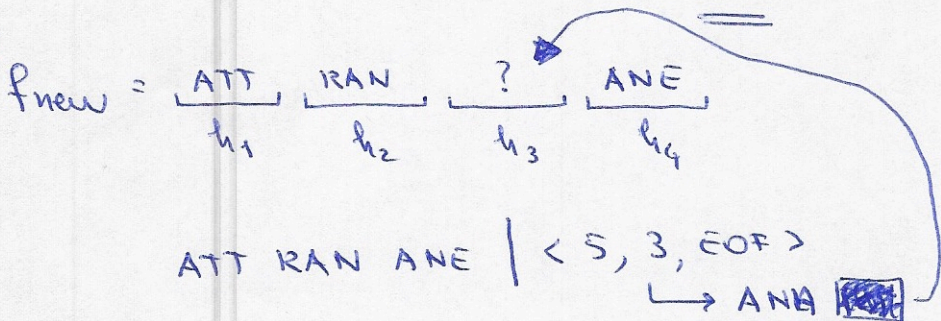


Other edges with the ~~same~~ present ones give the solution.

There are <sup>at least</sup> two minimum complements:

aabb, aaba, aba, ~~baaa~~ OR aaba, aba, aabb  
baaa

Q5



26. The idea is to interpret movies as "Terms" and users as "documents". So the solution consists of solving a soft-and with a lower-bound on the number of ~~terms~~ "Terms" in  $S$  which are contained in a "document".

So we build an inverted list, where we have a dictionary of movies and the pointing lists are on the users. Users get IDs so that they can be sorted.

Given  $S$ , we perform a soft-and over the pointing lists of the movies in  $S$ , and whenever we find a user shared by at least  $K$  lists, we return it.

---

Un altro modo di risolverlo era quello di costruire una rappresentazione con vettore binario per ogni utente e poi applicare LSH impostando soglia di similarità in modo da avere equal the  $\phi_s$  and at least  $K$   $\phi_s$  equal.