## ALGORITMI E STRUTTURE DATI STATISTICA PER I BIG DATA APPELLO DELL'11 FEBBRAIO 2022

In questo esame implementiamo un algoritmo che restituisce gli elementi di k-maggioranza; cioè che compaiono almeno n/k+1 volte nello stream. Chiaramente in ogni stream ci possono essere al più k-1 elementi di k-maggioranza. Nel caso lo stream abbia l < k-1 elementi di k-maggioranza, l'algoritmo deve restituire gli l elementi di k-maggioranza seguiti da un qualsiasi numero di elementi dello stream per un massimo di k-1 elementi in totale.

L'algoritmo per il caso k=2 può essere descritto nel modo seguente:

- (1) poni elmt = None e c = 0;
- (2) ogni volta che un nuovo elemento  $a_i$  dello stream arriva:
  - (a) se  $a_i = \text{elmt allora } c = c + 1;$
  - (b) se  $a_i \neq \text{elmt e } c = 0$  allora  $\text{elmt} = a_i \text{ e } c = 1$ ;
  - (c) se  $a_i \neq \text{elmt e } c > 0$  allora c = c 1;

Il problema vi chiede di implementare quindi la generalizzazione dell'algoritmo per valori di  $k \geq 2$ .

Ad esempio, supponiamo k = 3. Per lo stream:

abbiamo due elementi di k-maggioranza e precisamente 2 e 3.

Istruzioni per la consegna. Il codice da sviluppare consiste di una classe kmajority che offre i seguenti 3 metodi:

- (1) il costruttore  $\_$ init $\_$  che prende in input l'intero k;
- (2) il metodo addElement che aggiunge un elemento dello stream;
- (3) il metodo returnKMaj che restituisce la lista degli elementi di k-maggioranza Tutto il codice consegnato deve essere contenuto in un file con estensione .py che ha come nome il cognome dello studente scritto in minuscolo. Se il cognome contiene un apostrofo, uno spazio o un accento questi dovranno essere omessi.

La cartella che ha ricevuto contiene il pdf di questa traccia e il file driver.py che può essere usato per verificare il funzionamento della funzione progettata.