33. Hemos visto como usar reservoir sampling para muestreo de un stream del que no conocemos su longitud. Queremos extender este resultado a la estimación de valores de una función definida sobre el stream.

Tenemos un stream $s = a_1, a_2, \ldots, a_m$ formado por valores enteros en $[n] = \{0, \ldots, n-1\}$. Para $i \in [n]$, f_i denota la frecuencia de i en s. Queremos estimar el valor

$$g(s) = \sum_{i \in [n]} g(f_i),$$

donde g es una función con valores reales, con $g(0) = 0$.

El siguiente algoritmo combina la obtención de una muestra con un conteo parcial:

- Obtener x, una muestra (con distribución uniforme) sobre $[m]$
- $r = |\{i \geq x \mid a_i = a_x\}|$
- Return $m\,(g(r) - g(r-1))$

a) Demostrad que el algoritmo propuesto proporciona un estimador de la función g.

b) Proporcionad una implementación del algoritmo propuesto para un stream de datos del que no conocéis su longitud m.

a) Per veure que proporciona un estimador hem de comprovar que el resultat esperat és igual que el valor a estimar, en aquest cas g(s).
Esperança del resultat:
$E(m * (g(r) - g(r-1))) \rightarrow m * (E(g(r)) - E(g(r-1)))$

Tenint en compte que la esperança es pot calcular com el sumatori de les probabilitats per el valor, tenim:

$$m * \left(\sum_{i \epsilon [m]} \frac{1}{m} * g(r) - \sum_{i \epsilon [m]} \frac{1}{m} * g(r-1) \right)$$

Sabem que per un element que apareix f_i vegades calcularem tantes r com aparicions. És obvi veure que per la primera aparició $r = f_i$, per la segona $r = f_i - 1$ … fins a la última que serà $r = 1$. Per tant, per cada element diferent ens quedem amb:

$$g(f_i) + g(f_i - 1) + \ldots + g(0)$$

Per tant podem reescriure la fórmula anterior de la següent manera:

$$\sum_{i \epsilon [n]} \sum_{k=1}^{f_i} g(k) - \sum_{i \epsilon [n]} \sum_{k=1}^{f_i} g(k-1) \rightarrow \sum_{i \epsilon [n]} \sum_{k=1}^{f_i} g(k) - g(k-1)$$

El sumatori de dintre el podem simplificar si ens fixem en que podem simplificar tots els termes exceptuant $g(f_i) - g(0)$

$$\sum_{i \epsilon [n]} g(f_i) - g(0) \rightarrow \sum_{i \epsilon [n]} g(f_i) = g(s)$$

Hem trobat que es compleix la condició que buscavem, per tant podem concloure que l'algorisme és un estimador de g.

b)

```
aproximarG(stream s){
        m = 0;
        while not s.end(){
                a = s.read()
                ++m;
                amb probabilitat 1/m {
                        x = a;
                        r = 0;
                }
                if a == x : ++r;
        }
        On query, report : m*(g(r) - g(r-1))
}
```