Ampliació d'Algorismia: Aproximation, Parameterization and Streaming

QP 2022-2023: Sheet 3

29. Tenim un flux de dades que conté tots els nombres al conjunt $\{1, \ldots, n\}$ menys un. Dissenyeu un algorisme per trobar l'element que falta. El vostre algorisme pot llegir el flux només una vegada i fer servir $O(\log n)$ espai a memòria. L'algorisme ha de proporcionar la resposta correcta després de processar tota l'entrada.

30. Considereu el següent algorisme que processa un graf G amb n nodes i donat per un flux d'arestes.

    1. Selecciona una aresta (u, v) amb distribució uniforme sobre les arestes d'E.
    2. Selecciona amb distribució uniforme un vèrtex de $V \setminus \{u, v\}$
    3. Si (u, w) i v, w apareixen després de (u, v) en el flux, la sortida és $m(n-2)$, si no serà 0.

    (a) Proporcioneu una implementació com algorisme de streaming. El vostre algorisme només pot fer una pasada per el stream i utilitzar $O(n \log n)$ memoria.

    (b) Demostreu que l'algorisme proporciona una estimació del nombre de triangles al graph G.

31. Consider the following graph streaming algorithm

```
procedure PartTree(int n, stream s)
    F = ∅, x = 1
    while not s.end() do
        (u, v) = s.read()
        if x then
            if F ∪ {(u, v)} does not contain a cycle then
                F = F ∪ {(u, v)}
            else
                if F ∪ {(u, v)} contains an odd cycle then
                    x = 0
                end if
            end if
        end if
    end while
    On query, report x
end procedure
```

- Analyze the performance of PartTree.
- Show that the answer to a query is 1 iff and only if the already seen graph is 2-colorable.
- Modify the algorithm so that in addition, when the answer is 1, it provides a valid 2-coloring of G. Justify the efficiency and correctness of your proposal.

32. Considereu el següent algoritme de streaming que intenta resoldre el problema de trobar un matching amb pes màxim a un graf amb pesos a les arestes. Per aquest tipus de grafs l'algorisme té un paràmetre addicional $\gamma > 0$.

Per simplicitat al codi veurem una aresta $e = (u, v)$ com el conjunt $\{u, v\}$. Així podem escriure la condició $e \cap e' \neq \emptyset$ indicant si les arestes e i e' son incidents o no. Cada component del stream té dos valors indicant el primer el conjunt corresponent als vèrtexs de l'aresta i el segon el pes de aquesta aresta.

procedure GWM(int n, stream s, double γ)
  $M = \emptyset$
  while not $s.end()$ **do**
    $(e, w) = s.read()$
    $C = \{e' \in M \mid e \cap e' \neq \emptyset\}$
    $W = \sum_{e \in C} w(e)$
    if $w/W \geq (1 + \gamma)$ **then**
      $M = (M \cup \{e\}) \setminus C$
    end if
  end while
  On query, report M
end procedure

Per analitzar el rendiment de aquest algorisme diem que un aresta es un *supervivent* si forma part del conjunt M al finalitzar el tractament del stream. Cada supervivent $e \in M$ ha deixat un rastre de arestes mortes, afegides a M en algun moment i després eliminades. Aquest rastre es pot formalitzar com $T_e = C_1 \cup C_2 \cup \ldots$, on els C's es poden definir recursivament. $C_0 = \{e\}$, i

$$C_i = \cup_{e' \in C_{i-1}} \{\text{arestes eliminades de } M \text{ en afegir } e'\}.$$

Definim $T(M) = \cup_{e \in M} T_e$. Demostreu què:

(a) $w(T(M)) \leq w(M)/\gamma$

(b) $\mathsf{opt}(G) \leq (1 + \gamma)(w(T(M)) + 2w(M))$

(c) Quin és el valor de γ que proporciona el millor rendiment per GWM?

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.