CC4102 - Control 3

Prof. Gonzalo Navarro

16 de Noviembre de 2017

P1 (2.0 pt)

Un elemento x es α-mayoritario en $A[1,n]$ si aparece más de $\alpha \cdot n$ veces, para $0 < \alpha < 1$. Se desea usar un algoritmo tipo MonteCarlo para encontrar un elemento α-mayoritario en tiempo $O(n)$.

1. (1 pt) Diseñe un algoritmo que encuentre un elemento α-mayoritario o diga que no existe. Si existe un α-mayoritario, debe encontrarlo con una probabilidad de al menos α. Si no existe, debe decir que no existe, sin error.

2. (1 pt) Reduzca la probabilidad de error a un $\epsilon > 0$ dado, y dé el costo del algoritmo en términos de α, n, y ϵ.

P2 (2.0 pt)

Recuerde el problema de recubrir los vértices de un grafo $G = (V, E)$ cuando hay pesos $c : V \rightarrow R^+$ asociados a cada nodo. Se desea encontrar un $V' \subseteq V$ tal que toda arista de E tenga al menos una de las dos puntas en V', y minimizar $C = \sum_{v \in V'} c(v)$.

En clase se vio una 2-aproximación sencilla para el caso sin pesos ($c(v) = 1$) y una basada en programación lineal para el caso con pesos. Considere ahora la siguiente variante *aleatorizada* del método más sencillo, pero que funciona con pesos:

Algoritmo: *Partir de un $V' = \emptyset$. Ir tomando las aristas $e \in E$ en algún orden. Para cada $e = \{u, v\}$, con probabilidad $\frac{c(v)}{c(u)+c(v)}$ incluir u en V', sino incluir v en V'. Luego eliminar todas las otras aristas incidentes en el nodo incluido, y continuar eligiendo otro e hasta vaciar E.*

(Observe que u se incluye con probabilidad $c(v)/\ldots$, no $c(u)/\ldots$).

Demuestre que este algoritmo aleatorizado obtiene, en el caso esperado, una 2-aproximación. Para ello, considere las aristas e que el algoritmo procesa, metiendo una de sus dos puntas en V'. Sume la contribución esperada a C de la *arista* e, en vez de pensar en los *nodos* de esa arista. Compare con lo que debería hacer un algoritmo óptimo con e, de forma similar a como se hizo en clase para el caso $c() = 1$.

P3 (2.0 pt)

Considere una versión modificada del problema de la mochila, en la que se tienen n elementos cuyos pesos son números reales que suman m, y se dispone de m mochilas que pueden soportar peso hasta 1.0. Se desea maximizar el *número total de elementos* que se introducen en las mochilas, sin sobrepasar la capacidad de ninguna. Diseñe una 2-aproximación para este problema.

Tiempo: 2.0 horas   Con una hoja de apuntes   Responder en hojas separadas