CC4102 - Examen

Prof. Gonzalo Navarro

5 de Enero de 2021

P1 (1.5 pt)

El Profesor Locovich está empeñado en determinar cuál es el último piso de un edificio desde el que puede tirarse un huevo de avestruz sin romperse. El edificio tiene n pisos, pero dado el alto costo de los huevos de avestruz, el Profesor sólo dispone de k huevos. Debe usted diseñar un algoritmo para resolver el problema. El costo de su algoritmo es la cantidad de veces que necesita tirar el huevo para responder.

1. (0.5pt) Demuestre que si $k = 1$, la complejidad del problema es n.
2. (0.5pt) Encuentre un algoritmo de costo $O(\sqrt{n})$ para $k = 2$.
3. (0.5pt) Demuestre que, con $k = 2$, la complejidad del problema es $\Omega(\sqrt{n})$.

P2 (2.5 pt)

Un determinado algoritmo de compresión sobre un texto $T[1..n]$ funciona como sigue:

1. Recolecta todos los pares distintos $ab = T[i]T[i+1]$ que aparecen en el texto, con la cantidad de veces que aparece cada uno.
2. Selecciona el par más frecuente, digamos ab.
3. Reemplaza todas las ocurrencias de ab en T por un nuevo símbolo A (note que si $a = b$, el par aa podría reemplazarse sólo una vez cada dos en $aaaaaa$, pero puede ignorar esto).
4. Anota la regla $A \to ab$.
5. Actualiza los nuevos pares para considerar el reemplazo de cada ab por A, es decir, si donde decía $cabd$ ahora dice cAd, entonces ha desaparecido una ocurrencia de los pares ca y bd y ha aparecido una de cA y Ad.
6. Si queda en T algún par que se repite, vuelve al punto 2.
7. El output es la secuencia final más las reglas que se produjeron.

Se pide:

1. (0.5pt) Defina las estructuras de datos necesarias para poder realizar el paso 2, y los pasos 3 y 5 por cada ocurrencia de ab, en tiempo $O(\log n)$. Indique cuánto espacio ocupan sus estructuras y si su tiempo es de peor caso o esperado.
2. (0.5pt) Dado el punto anterior, demuestre que el algoritmo completo funciona en tiempo $O(n \log n)$.
3. (0.5pt) Demuestre que los pares cuya frecuencia se modifica, y los que se generan, nunca superan la frecuencia del que se está reemplazando.
4. (0.5pt) Usando el hecho anterior, modifique sus estructuras de datos para reducir la complejidad total a $O(n)$.
5. (0.5pt) Entendiendo que el alfabeto original de T es $\{1, 2, \ldots, \sigma\}$ y que usted puede numerar los símbolos nuevos A como desee, diseñe y analice un algoritmo de peor caso $O(n)$ para reconstruir T a partir de la secuencia final y las reglas.

P3 (1.5 pt)

Sea S un conjunto de n elementos. Sean $S_1, \ldots, S_k$ subconjuntos de S, cada uno de ellos de tamaño exactamente r. Suponga que se cumple $k \cdot 2^{-r} \leq 1/4$ (es decir no pueden ser demasiados conjuntos muy chicos). Se desea colorear los elementos de S rojo o azul, de modo que todo S_i contenga elementos de ambos colores.

1. (0.5pt) Proponga un algoritmo tipo Monte Carlo que encuentre un coloreo válido con probabilidad al menos $1/2$, y analice su costo de peor caso.
2. (0.5pt) Reduzca la probabilidad del algoritmo anterior a $1/2^t$ para cualquier $t > 0$ dado, y analice su costo de peor caso.
3. (0.5pt) Proponga un algoritmo tipo Las Vegas y analice su costo promedio.

P4 (2.0 pt)

En clase se vio una 2-aproximación sencilla para el recubrimiento de vértices sin pesos. Considere ahora la siguiente variante aleatorizada del método: cada vez que considera una arista (u, v), elige u con probabilidad $1/2$ y elige v con probabilidad $1/2$, independientemente. En cada caso, si elige un nodo, elimina todas las otras aristas que inciden en él.

Sea S_i el conjunto de nodos elegidos hasta el paso i y sea OPT un conjunto óptimo.

1. (0.5pt) Demuestre que, en todo momento, por cada arista que no ha sido aún eliminada, el algoritmo óptimo deberá elegir al menos uno de sus extremos.
2. (1pt) Demuestre que lo anterior implica que para todo i vale $E(|S_i \cap OPT|) \geq E(|S_i \setminus OPT|)$.
3. (0.5pt) Demuestre que lo anterior implica que el tamaño esperado del conjunto de nodos que entrega el algoritmo es una 2-aproximación.