CC4102 - Examen

Prof. Gonzalo Navarro

18 de Julio de 2023

P1 (2 pt)

Un número es *mayoritario* en un arreglo de n elementos si aparece más de $n/2$ veces. Por ejemplo 3 es mayoritario en $\{1,3,3,5,3,3,2,3,3,4,3,4,1,3,1\}$. # Es un modelo de comparación

1. (1 pt) Demuestre que cualquier algoritmo (determinístico) que resuelva el problema debe examinar todas las celdas del arreglo, usando el argumento del adversario.
2. (1 pt) Diseñe un algoritmo de tipo Las Vegas que, en caso de existir un elemento mayoritario, lo encuentre en tiempo esperado $O(n)$. Si exigimos que el algoritmo tome $O(kn)$ en el peor caso, ¿cuál es la probabilidad de error?

Solución: # 1. Se tienen los n elementos, el algoritmo elige un x e y, obteniendo la frecuencia de ambos, si x=y suma 2 a dicho numero, si x!=y suma 1 a cada numero. El adversario se encarga de que cada numero tenga la misma frecuencia, obligando al algoritmo a comparar todas las celdas para definir que numero es mayoritario.

Para la primera parte, el adversario elige dos valores distintos a y b, y va entregando alternadamente a y b en cada nueva celda que se accede. Sólo en la última celda se define cuál de los dos es el mayoritario, el algoritmo no puede responder antes.

En la segunda, elijo al azar un elemento y pago $O(n)$ para ver si es mayoritario. Si no lo era, elijo al azar otro elemento, y así, hasta dar con uno. La probabilidad de realizar más de k pasadas es $1/2^k$, por lo que el número esperado de pasadas es 2. Si nos tenemos que detener luego de k pasadas, la probabilidad de error es $1/2^k$. # 2. Se cuenta cuantas veces el elemento elegido está en el arreglo para ver si mayor a n/2 -> O(n).
Como la frecuencia de un elemento mayoritario es n/2. La probabilidad de seleccionar dicho elemento en una sola pasada es 1/2, por ende la prob. de realizar más de k pasadas para encontrarlo es 1/2^k

P2 (2 pt)

Un *conjunto independiente* de un grafo $G=(V,E)$ con $|V|=n$ y $|E|=m$ es un subconjunto V' de V sin aristas de E entre elementos de V'. El problema de encontrar un conjunto independiente de cierto tamaño es NP-completo. # V = Conjunto de vértices & E = conjunto de aristas

Considere el siguiente algoritmo aleatorizado para encontrar un conjunto V':

1. Partiendo de un V' vacío, se agrega cada vértice de V con una cierta probabilidad p.
2. Para cada arista de E con ambos vértices en V', se saca uno de los dos vértices (elegido al azar) de V'.

Se pide (son varias cosas pero cortas):

1. Muestre que el resultado del algoritmo es un conjunto independiente.
2. Calcule la cantidad esperada v de vértices en V' luego del paso 1.
3. Calcule la cantidad esperada e de aristas entre vértices de V' luego del paso 1.
4. Calcule el tamaño esperado V' luego del paso 2.

5. Maximice el tamaño esperado en términos de p, indicando el tamaño esperado máximo M que puede conseguir, y el valor de p para conseguirlo.

6. En base al resultado anterior, demuestre que todo grafo tiene un conjunto independiente de tamaño al menos M.

Solución:

1. En el paso 2 se elimina de V' un vértice de cada arista que existiera entre vértices de V'. Al terminar no puede quedar ninguno porque no se reintroducen vértices, sólo se eliminan.

2. Cada una de las n aristas de V se inserta con probabilidad p, por lo que $v = pn$.

3. La probabilidad de que una arista de E esté en V' es de que sus dos puntas hayan sido elegidas, p^2, por lo tanto $e = p^2 m$.

4. Por linealidad de la esperanza, el tamaño esperado de V' es $v - e = pn - p^2 m$.

5. Derivando con respecto a p, obtengo que para $p = n/(2m)$ obtengo un tamaño esperado de $M = n^2/(4m)$.

6. Dado que M es el tamaño esperado de un conjunto independiente que produzco, deben existir conjuntos independientes mayores que M.

Como M es el tamaño esperado de un cjto independiente, implica que la prob de que el tamaño en alguna otra ejecucion sea menor que M es baja. Y por consiguiente existirán conjuntos indp. mayores que M.

P3 (2 pt)

Se tiene un árbol escrito en un arreglo $A[1..n]$, donde cada elemento $A[i] = j$ indica que el padre de i es j. La raíz r indica $A[r] = 0$.

1. (1 pt) Diseñe un algoritmo PRAM para calcular la profundidad de cada nodo (distancia a la raíz) en un arreglo $P[1..n]$, en tiempo $O(\log n)$. Analice $T(n)$, $W(n)$, speedup, eficiencia, y número óptimo de procesadores. ¿Qué tipo de modelo usó? (EREW, CREW, CRCW).

2. (1 pt) Lo mismo, ahora para calcular la altura de cada nodo (distancia a la hoja más profunda que desciende de él). Considere un modelo CRCW donde, si hay varias escrituras concurrentes, la celda se queda con el máximo valor escrito.

Solución:

El primero es un CREW muy parecido a list ranking. El segundo puede hacer una pasada para anotar quiénes son hojas (cada nodo le anota a su padre que no lo es) y luego las hojas tienen altura 0 y les escriben a su padre altura+1, duplican links, todos escriben altura+2 (si es mayor a la altura que ya tenia), etc.

Tiempo: 3 horas | Con tres hojas de apuntes