

IIC2283 — Diseño y Análisis de Algoritmos

Ayudantía 3

Juan Navarro

Problema 1

Problema 3 ayudantía anterior.

Problema 2

1. Se considera que una moneda falsa es más liviana o más pesada que una auténtica. Luego, dadas n monedas, donde a lo más una es falsa, se busca determinar si efectivamente hay una moneda falsa, y en ese caso indicar cual es, y si es más liviana o más pesada que las otras. Para esto, se dispone de una balanza, con la cual se pueden comparar dos conjuntos de monedas A y B , obteniendo uno de tres resultados: $w(A) = w(B)$, $w(A) < w(B)$, o $w(B) < w(A)$.

Determine una cota inferior para la cantidad de usos de la balanza en el peor caso.

Idea de solución El problema se puede modelar con un árbol de decisión ternario. Existen $1 + 2n$ outputs posibles: 1 que no hay una moneda falsa, y 2 respuestas posibles por cada moneda (más liviana, o más pesada). Luego, todo árbol tiene altura mayor o igual a $\lceil \log_3(1 + 2n) \rceil$, y por lo tanto en el peor caso se necesita al menos esa cantidad de usos de la balanza.

2. Generalice el resultado anterior, pero con a lo más k monedas falsas, y sin considerar si son más pesadas o más livianas. Se puede asumir que $k \leq n/2 - \sqrt{n}$.

Idea de solución Solo cambia la cantidad de hojas. Si existen ℓ monedas falsas, entonces hay $\binom{n}{\ell}$ respuestas posibles. Luego, la cantidad de hojas corresponde a:

$$\sum_{i=0}^k \binom{n}{i}.$$

Se considerarán dos cotas posibles. Primero que:

$$\sum_{i=0}^k \binom{n}{i} > \sum_{i=0}^k \binom{k}{i} = 2^k,$$

obteniendo que la altura es $\Omega(k)$.

También, se puede acotar que $\sum_{i=0}^k \binom{n}{i} > \binom{n}{k} > (\frac{n}{k})^k$, y luego obtener que la altura es al menos $k(\log_3(n/k))$.

La demostración de $\binom{n}{k} > (\frac{n}{k})^k$ se puede obtener considerando que si $0 < i < k < n$, entonces:

$$\begin{aligned}
\frac{i}{n} &< \frac{i}{k} \\
1 - \frac{i}{k} &< 1 - \frac{i}{n} \\
\frac{k-i}{k} &< \frac{n-i}{n} \\
\frac{n}{k} &< \frac{n-i}{k-i}
\end{aligned}$$

y luego que:

$$\begin{aligned}
\binom{n}{k} &= \frac{n(n-1)(n-2)\cdots(n-k+1)}{k!} \\
&= \frac{n}{k} \cdot \frac{n-1}{k-1} \cdots \frac{n-(k-1)}{k-(k-1)} \\
&\geq \frac{n}{k} \cdot \frac{n}{k} \cdots \frac{n}{k} \\
&= \left(\frac{n}{k}\right)^k.
\end{aligned}$$

Problema 3

En este problema se asumirá que determinar si una lista de n números reales contiene elementos repetidos requiere $\Omega(n \log n)$ comparaciones.

Encuentre cotas inferiores para los siguientes problemas:

1. Dados n números reales, encontrar el par más cercano.

Solución Reducción desde elementos repetidos. Se mantiene la misma entrada, y luego se revisa la distancia entre los elementos más cercanos. Si es cero, entonces hay elementos repetidos, y si es mayor a cero, entonces no los hay. Luego la cota inferior es $\Omega(n \log n)$.

2. Dados n puntos en el plano, determinar el MST de ellos bajo distancia euclidiana.

Idea de solución Similar al anterior.

3. Dados n números enteros, y n cajas con signos $<$, $\leq$, $\geq$ o $>$ entre ellas, asignar los números de forma que se satisfagan todas las desigualdades.

Idea de solución Reducción desde ordenación, usando signos $\leq$. La cota encontrada es de $\Omega(n \log n)$.

Problema 4

Definimos un *linear decision tree* (LDT) de dimensión n como un árbol ternario, donde cada nodo interno está etiquetado con un polinomio lineal con n indeterminadas, cada hoja con un 0 o un 1, y las ramas que salen de un nodo con los signos $-$, 0 y $+$.

El árbol recibe como entrada un vector $\vec{x} = \{x_1, \dots, x_n\}$, y luego construye un camino desde la raíz hasta una hoja, en base a evaluar el polinomio correspondiente en cada nodo. Comenzando desde la raíz, si el resultado de evaluar el polinomio es negativo, se continúa por la rama $-$, si es cero por la rama 0, y si

es positivo, por la rama +, hasta alcanzar una hoja. Finalmente, el árbol acepta el vector si se alcanza una hoja con etiqueta 1, y rechaza si se llega a una hoja con etiqueta 0.

El funcionamiento de un LDT puede interpretarse geoméricamente, considerando que cada nodo interno define un hiperplano que divide el espacio en tres conjuntos convexos: el hiperplano mismo, y los dos semiespacios, que corresponden a las tres ramas del árbol.

Parte 1 Muestre que la intersección de conjuntos convexos es convexa.

Idea de solución Dados dos puntos arbitrarios en la intersección, entonces pertenecen a todos los conjuntos intersectados. Luego, los puntos en la recta entre ellos están contenidos en cada uno de los conjuntos, y por lo también en la intersección de ellos.

Dado un nodo v , denotaremos por $P(v)$ al conjunto de puntos que pasan por el nodo.

Parte 2 Demuestre que para todo nodo v , $P(v)$ es convexo y conexo.

Idea de solución El conjunto $P(v)$ está definido como la intersección de los conjuntos definidos por los hiperplanos, de acuerdo al camino seguido desde la raíz, por lo que es convexo. Además todo conjunto convexo es conexo.

Parte 3 Muestre que dos puntos llegan a la misma hoja, si y solo si pertenecen a la misma componente conexa.

Idea de solución Dos puntos llegan a la misma hoja, si y solo si cumplen las mismas relaciones con todos los hiperplanos del camino, lo que es equivalente a que pertenecen a la intersección de todos los conjuntos correspondientes, y por lo tanto pertenecen al mismo conjunto.

Si el árbol calcula una función F , se denomina $\#F_0$ al número de componentes conexas que cubren $F^{-1}(0)$, y $\#F_1$ al número correspondiente a $F^{-1}(1)$.

Parte 4 Demuestre que todo árbol que calcula F , tiene altura al menos $\log_3(\#F_0 + \#F_1)$.

Idea de solución Por definición existen al menos $\#F_0 + \#F_1$ componentes conexas, y además a cada una le corresponde una hoja distinta del árbol. Como es un árbol ternario, entonces tiene altura al menos $\log_3(\#F_0 + \#F_1)$.

Parte 5 Demuestre que todo LDT que decide si una lista de largo n tiene elementos repetidos tiene altura superior a $(n/2) \log_3(n/2)$.

Solución Se considera un vector sin elementos repetidos $p = \{p_1, p_2, \dots, p_n\}$, por lo que $F(p) = 1$. Luego, el punto $p' = \{p_2, p_1, \dots, p_n\}$ tampoco tiene elementos repetidos, y cumple que $F(p') = 1$.

Luego, se considera el hiperplano definido por $H = x_1 - x_2$, donde se observa que $H(p) > 0$ y $H(p') < 0$, o bien $H(p) < 0$ y $H(p') > 0$. Por el teorema del valor intermedio, se tiene que la recta entre p y p' debe cruzar H . Por lo anterior, para algún $0 < \lambda < 1$, se tiene que $H(\lambda p + (1 - \lambda)p') = 0$. Si denominamos $z = (z_1, \dots, z_n)$ a ese punto entonces concluimos que $F(p) = 1 = F(p')$, pero $F(z) = 0$. Luego, como los conjuntos son convexos, entonces necesariamente p y p' pertenecen a componentes distintas.

Siguiendo el argumento anterior, cada permutación posible de p pertenece a un conjunto convexo distinto, que corresponde a una hoja distinta con etiqueta 1, lo que implica que $\#F_1 \geq n! > (n/2)^{(n/2)}$. Luego, por los resultados anteriores, la altura del árbol es al menos $(n/2) \log_3(n/2)$.

Parte 6 Muestre que los árboles de comparación son casos particulares de LDTs.

Idea de solución Un árbol de comparaciones es un LDT donde en cada nodo interno, el polinomio es de la forma $x_i - x_j$, y los subárboles para $-$ e $=$ son iguales.

Parte 7 Demuestre que todo algoritmo basado en comparaciones para ese problema realiza $\Omega(n \log n)$ comparaciones.

Idea de solución A cada algoritmo basado en comparaciones le corresponde un árbol de comparaciones. Si existe un árbol de comparaciones con altura $o(n \log n)$ para el problema, entonces se podría construir un LDT de igual altura, obteniendo una contradicción.