

Auxiliar 1 - “Cotas inferiores: Adversario”

Profesores: Pablo Barceló
Gonzalo Navarro
Auxiliar: ~~Manuel~~ Ariel Cáceres Reyes

19 de marzo del 2018

P1. Mediana

Considere el problema de encontrar la *mediana* de un arreglo de n elementos distintos (con n impar).

- a) Muestre que se necesitan n accesos al arreglo para responder la mediana
- b) Muestre que se necesitan al menos $n - 1$ comparaciones entre elementos
- c) Muestre que se necesitan al menos $\sim \frac{3n}{2}$ comparaciones entre elementos

P2. Grado 3

Muestre que determinar si un grafo no dirigido posee o no un vértice de grado 3 toma, en el peor caso, $\binom{n}{2} - n$ consultas del tipo “¿existe un arco entre los vértices u y v ?”, si la cantidad total de vértices es n .

P3. Mezclar 2 listas ordenadas

Considere el problema de mezclar 2 listas ordenadas de igual tamaño, $X[1, \dots, n], Y[1, \dots, n]$ de modo que el resultado final este también ordenado. En este problema estaremos interesados en las preguntas del estilo: ¿ $X[i] < Y[j]$?

- a) Muestre que el problema es $\mathcal{O}(2n - 1)$
- b) Muestre que el problema no puede ser resuelto en menos de $2n - 1$ comparaciones

“Sometimes, someone you thought was your adversary is your ally.”
Erica Durance

P1. Mediana

- a) Por contradicción, si un algoritmo encuentra la mediana en menos de n accesos significa que no accede a alguno de los elementos del arreglo. El adversario puede en este caso elegir el valor de este elemento no accedido a su elección de modo que la elección de mediana del algoritmo sea incorrecta.
- b) Consideremos el grafo G cuyos vértices representan a los elementos del arreglo y las aristas representan las comparaciones realizadas por el algoritmo. Si un algoritmo hace menos de $n - 1$ comparaciones significa que G no es conexo. Sea m el vértice que representa la mediana escogida por el algoritmo, el cual se encuentra en una de las componentes conexas de G . El adversario puede entonces, tomar otra componente conexa y hacer que todos esos elementos sean menores o mayores a m que en alguno de los dos casos hace que m sea una respuesta a mediana incorrecta.
- c) Consideremos ahora un grafo dirigido D , cuyos vértices son los elementos del arreglo y existe una arista $u \rightarrow v$ si se le respondió al algoritmo que el elemento u era mayor que el v . Sea m el vértice que representa a la mediana, al finalizar las consultas se debe cumplir que en D todos los vértices tienen un camino dirigido hacia m o desde m (si existe un elemento que no cumpla esta condición el adversario puede establecer su valor arbitrariamente mayor o menor al de m y hacer que m sea una respuesta de mediana incorrecta). Nombremos comparaciones cruciales a todas aquellas que se encuentran en un camino desde o hacia m . Por la condición anterior sabemos que existen al menos $n - 1$ comparaciones cruciales.

Notemos que el resto de comparaciones (no cruciales) se hace entre elementos mayores a m y elementos menores a m . A continuación mostraremos un adversario que genera $\sim \frac{n}{2}$ comparaciones no cruciales, con lo que habremos terminado la demostración.

El adversario mantiene 3 conjuntos; A el de los elementos que nunca han sido comparados, B el de los elementos menores a la mediana y C el de los elementos mayores a la mediana. Inicialmente comienza con todos los elementos en A , siendo B y C vacíos.

- Si se realiza una comparación entre dos elementos de A se escoge uno mayor que la mediana (y se pone en C) y el otro menor (y se pone en B).
- Si se realiza una comparación entre dos elementos de B o entre dos elementos de C , se elige una manera de responder consistente con las respuestas ya dadas.
- Si se realiza una comparación entre un elemento de A y uno de B o uno de C , se responde de manera que la comparación sea no crucial.
- Las otras comparaciones se responden de acuerdo al nombre de los conjuntos.

Para finalizar hay que notar que para que el algoritmo haya comparado todos los elementos se necesitan al menos $\frac{n-1}{2}$ comparaciones no cruciales.

“Sometimes, someone you thought was your adversary is your ally.”

Erica Durance

P2. Grado 3

En este caso nuestro adversario mantendrá un grafo G que contiene las aristas que aún no han sido preguntadas por el algoritmo.

Inicialmente el adversario tiene un grafo completo $G = V^2$. Cada vez que el algoritmo le pregunta por la existencia de una arista $\{u, v\}$ el adversario setea $G \leftarrow G - \{u, v\}$, en caso que G no tenga un vértice de grado al menos 3 el adversario responde que la arista consultada es parte del grafo; en cambio, si G tiene un vértice de grado al menos 3 el adversario responde que la arista consultada NO es parte del grafo.

Veamos primero que el algoritmo necesita que el adversario le responda afirmativamente acerca de la existencia de una arista. Pues si nunca le responde afirmativamente significa que en G existe un vértice que tiene 3 aristas incidentes, es decir, 3 aristas que el algoritmo no ha consultado si pertenecen o no al grafo. Por lo tanto existen dos inputs posibles, consistentes con las respuestas del adversario, uno que tiene un vértice de grado 3 y otro que no lo tiene.

Veamos ahora que para que el adversario responda afirmativo a una pregunta se necesitan al menos $\binom{n}{2} - n$ consultas, con lo que finalizaremos esta demostración.

En el momento en que el adversario responde por primera vez afirmativamente, todos los vértices de G tienen grado ≤ 2 y por lo tanto G tiene a lo más n aristas. Finalmente, como cada arista que no está en G fue removida por una consulta anterior que se le hizo al adversario y como G comenzó con $\binom{n}{2}$ aristas, el algoritmo hizo al menos $\binom{n}{2} - n$ consultas.

P3. Mezclar 2 listas ordenadas

- El algoritmo de “merge” ocupado en “MergeSort” realiza en el peor caso $2n - 1$ comparaciones (una por cada elemento que pone en la lista de los ordenados). ⚡
- Por $\Rightarrow \Leftarrow$ existe un algoritmo A que realiza $\leq 2n - 2$ comparaciones.

El adversario 🐱 construye el siguiente input:

- $X \rightarrow$ lista cuyos elementos son $x_i = 2i - 1$
- $Y \rightarrow$ lista cuyos elementos son $y_i = 2i$

Si ejecutamos el algoritmo A sobre el input anterior encontraremos un i^* tal que x_{i^*} no fue comparado con ambos y_{i^*-1} e y_{i^*} (si todos los i lo cumplieran se habrían hecho más de $2n - 2$ comparaciones).

Supongamos sin mucha pérdida de generalidad que solo no fue comparado con y_{i^*} , entonces en el siguiente input:

“Sometimes, someone you thought was your adversary is your ally.”

Erica Durance

- $X \rightarrow$ lista cuyos elementos son $x_i = 2i - 1$, excepto $x_{i^*} = 2i^*$
- $Y \rightarrow$ lista cuyos elementos son $y_i = 2i$, excepto $y_{i^*} = 2i^* - 1$

El algoritmo recibe exactamente las mismas respuestas a las mismas preguntas que se hizo sobre el input anterior (pues el orden relativo de los elementos no se altera y no se compara x_{i^*} con y_{i^*}). Por lo tanto tengo 2 inputs válidos de listas para los cuales A retorna el mismo resultado lo que es una $\Rightarrow \Leftarrow$.

“Sometimes, someone you thought was your adversary is your ally.”
Erica Durance