Ejercicios: Final

gabriel.spranger@utec.edu.pe

December 22, 2020

1 Análisis Amortizado

1.1 Método de Agregación

Ejercicio 1. Suponga que en el problema de operaciones en pila incluimos una nueva operación MULTIPUSH, que pone $k \leq n$ elementos en la pila. ¿En esa situación, al hacer n operaciones, el costo amortizado de $O(1)$ por operación continúa valiendo? Suponga que ponemos la restricción de que el tamaño de la pila no puede sobrepasar n (el número de operaciones). ¿En esa situación el costo amortizado de $O(1)$ continúa valiendo?

El upper bound de $O(1)$ ya no sería correcto. Esto se debe a que podríamos hacer una serie de llamadas a MULTIPUSH(K) (*pushiamos* k elementos al stack) y MULTIPOP(K) (eliminamos k elementos del stack). Al usar el método de agregación, el costo amortizado de estas llamadas sería $O(\mathrm{k})$, ya que tendríamos:

$$\frac{n * O(k)}{n} = O(k)$$

Donde n es el número de llamadas a MULTIPUSH(K) y MULTIPOP(K).

Ejercicio 2. Muestre que si una operación DECREMENT es incluida en el problema del contador binario, n operaciones pueden costar $\Theta(nk)$.

Esto significa que ahora tenemos dos operaciones: INCREMENT y DECREMENT. Una serie de operaciones puede costar $\Theta(nk)$ en el siguiente caso. Imaginemos que nuestro string de bits tiene solo 0s. Si aplicamos la operación DECREMENT, habrá overflow y todo el string de bits ahora tendrá solo 1s. Esto quiere decir que se cambiaron todos los k bits del string, lo cuál tomará $\Theta(k)$. Luego, como tenemos un string que solo tiene 1s, si aplicamos la operación INCREMENT, ocurrirá otro overflow, el cual causará que todos los k bits del string cambien a 0, lo cual cuesta $\Theta(k)$. Si hacemos n operaciones de este tipo, el tiempo total sería:

$$n * \Theta(k) = \Theta(nk)$$

Ejercicio 4. Suponga que hacemos una secuencia de n operaciones en una estructura de datos en la cual la i-ésima operación cuesta i si i es una potencia de 2, y cuesta 1 en caso contrário. Use análisis agregado para determinar el costo amortizado por operación.

$$
\begin{aligned}
&\sum_{i=1}^{n} c_i \\
&= \sum_{i=1}^{\lceil \lg n \rceil} 2^i + \sum_{i \leq n (\text{no es potencia de 2})} 1 \\
&= \sum_{i=0}^{\lceil \lg n \rceil + 1} 2^i + \sum_{i \leq n (\text{no es potencia de 2})} 1 \\
&= 2^{\lceil \lg n \rceil + 1} - 1 + \sum_{i \leq n (\text{no es potencia de 2})} 1 \\
&\leq 2^{\lceil \lg n \rceil + 1} - 1 + n \\
&\leq 2n - 1 + n \\
&\leq 4n - 1 \\
&\leq 5n = O(n)
\end{aligned}
$$

Como estamos usando el análisis agregado bastaría hacer lo siguiente para hallar el costo amortizado por operación:

$$\frac{O(n)}{n} = O(1)$$

1.2 Método de Recargas

17.2-1
Suppose we perform a sequence of stack operations on a stack whose size never exceeds k. After every k operations, we make a copy of the entire stack for backup purposes. Show that the cost of n stack operations, including copying the stack, is $O(n)$ by assigning suitable amortized costs to the various stack operations.

Por cada operación en el stack, cobramos el doble (dos veces). Una vez por el costo real de la operación y otra para pagar la copia de ese elemento luego cuando se copie todo el stack. Como siempre hay k operaciones entre cada copia del stack, siempre cobraremos de más, lo cual asegura un upper bound. Como el costo amortizado de cada operación es constante, tenemos que el costo de n operaciones sería $O(n)$.

_17.2-2_
Redo Exercise 17.1-3 using an accounting method of analysis.

El ejercicio 17.1-3 sería el Ejercicio 4.

Le asignamos un costo de 3 a cada operación. Mostraremos con inducción que este costo amortizado constante, garantiza un upper bound sobre las n operaciones.

La primera operación cuesta 1 y como cobramos 3, tenemos 2 de crédito. Ahora imaginen que acabamos de llevar a cabo la 2^i-ésima operación, por hipótesis de inducción, tenemos un crédito acumulado no negativo. Ahora, cada una de las $2^i - 1$ operaciones que le siguen, tienen un costo real de 1 y como cobramos 3 por cada una, tendremos un crédito acumulado de $2(2^i-1) = 2^{i+1}-2$. Ahora la operación que sigue es la 2^{i+1}-ésima, por la cual cobramos 3, entonces nuestro crédito acumulado sería $2^{i+1} - 2 + 3 = 2^{i+1} + 1$ y como la 2^{i+1}-ésima operación cuesta 2^{i+1}, nos sobrará $(2^{i+1} + 1)$-(2^{i+1})=1 de crédito, es decir, nos quedará un crédito no negativo. Por lo tanto, para cualquier operación, tendremos crédito no negativo. Como el costo amortizado de cada operación es $3 = O(1)$ (constante), un upper bound sobre n operaciones de este tipo sería $O(n)$.

1.3 Método Potencial

Revisar la sección 17.3 del Cormen.

2 Programación Dinámica

Ver asesorías anteriores.

3 Reducciones

Ejercicio 8. Responda verdadero o falso. En cada caso justifique su respuesta.

(a) Suponga que $X \preceq_P Y$. Si X es NP-completo entonces Y también lo es.

(b) Suponga que $X \preceq_P Y$. Si Y es NP-completo y X es NP entonces X es NP-completo.

(c) Suponga que $X \preceq_P Y$. X e Y no pueden ser ambos NP-completos.

(d) Suponga que $X \preceq_P Y$. Si X está en P, entonces Y es en P.

(e) Si $P \neq NP$ entonces NP-completo =P

(f) Si $P \neq NP$ entonces NP-difícil = NP-completo

(g) Si $L \in NP$ entonces no existe algoritmo polinomial para L.

(h) Si $L \in NP$ y L es NP-difícil entonces L es NP-completo

(a) Depende de Y. Si $Y \in NP$, entonces, entonces **Verdadero**, sino, **Falso**, ya que $Y \in$ NP-Difícil.

(b) No necesariamente, ya que cualquier NP se puede reducir polinomialmente a un NP-Completo, pero esto no garantiza que $X \in$ NP-Completo. Es decir, **Falso**.

(c) Por definición de NP-Completo, **Falso**.

(d) No necesariamente porque un P se puede reducir a un NP. Es decir, **Falso**.

(e) **Falso**.

(f) **Falso**.

(g) Si $L \in$ P, entonces **Falso**, sino, entonces **Verdadero**. Acuérdense que P $\subseteq$ NP.

(h) **Verdadero**. Ver el gráfico del frame 1 del Jamboard.

Problema CICLO-HAM. Dado un grafo G, decidir si G tiene un ciclo que pasa por todos vértices del grafo.

Problema TSP. Dado un grafo G con pesos en las aristas y un número W, decidir si G tiene un ciclo hamiltoniano con peso menor o igual a W.

Problema CAMINO-HAM. Dado un grafo G, decidir si G tiene un camino (simple) que pasa por todos los vértices del grafo.

Problema CUADRADO. Dado un número natural n, decidir si existe un número natural x tal que $x^2 = n$.

Problema EC-2-GRADO. Dados tres números naturales a, b, c, decidir si existe un número natural x tal que $ax^2 + bx + c = 0$.

Problema COMPUESTO. Dado un número natural $n > 1$, decidir si es un número compuesto.

Problema FACTOR. Dados dos números naturales n y m, decidir si existe un número natural p tal que $1 < p < m$ y p divide a n.

Problema SUBSET-SUM. Dado un conjunto $\{a_1, a_2, \ldots, a_n\}$ de números naturales y un número c, decidir si existe un subconjunto K de $\{1, 2, \ldots, n\}$ tal que $\sum_{i \in K} a_i = c$

Problema MOCHILA-VALIOSA. Dados dos conjuntos $\{p_1, p_2, \ldots, p_n\}$, $\{v_1, v_2, \ldots, v_n\}$ de números naturales y dos números naturales c y d, decidir si existe un subconjunto K de $\{1, 2, \ldots, n\}$ tal que $\sum_{i \in K} p_i \leq c$ y $\sum_{i \in K} v_i \geq d$

Problema CAMINO-HAM-S. Dado un grafo G y un vértice s en G, decidir si G tiene un camino (simple) que pasa por todos los vértices del grafo y comienza en s.

Problema CAMINO-LARGO. Dado un grafo G y un número k, decidir si G tiene un camino (simple) con por lo menos k aristas

Ejercicio 4. Sabiendo que COMPUESTO es NP-completo, muestre que FACTOR es NP-completo

1. FACTOR $\in$ NP
2. COMPUESTO $\leq_p$ FACTOR

1. Para construir el certificador B, el certificado tendría a los números n, m y p. B evaluaría que 1 < p < m y que n%p == 0. Esto se puede hacer en tiempo polinomial.

2. Queremos usar la caja negra que resuelve a FACTOR para resolver COMPUESTO. Compuesto retorna SÍ si el "n" es compuesto, es decir, si n tiene otro divisor aparte de n y 1, y retorna NO caso contrario. Por otro lado, FACTOR, recibe un n y m y retorna SÍ si existe un número natural p, tal que 1 < p < m y p divide a n. Por lo tanto, bastaría hacer que el n y m que recibe FACTOR, sea igual al n que recibe COMPUESTO. Si con esos parámetros, FACTOR retorna sí, significa que existe un p que no es ni 1 ni "n" tal que p divide a n, si dicho p existe, significa que n es compuesto, ya que existe otro número aparte de 1 y n que divide a n, ese número es p.

Ejercicio 5. Sabiendo que SUBSET-SUM es NP-completo, muestre que MOCHILA-VALIOSA es NP-completo

1. MOCHILA-VALIOSA $\in$ NP
2. SUBSET-SUM $\leq_p$ MOCHILA-VALIOSA

1. El certificado del certificador B consta de P = {p_1, p_2, ..., p_n} y V = {v_1, v_2, ..., v_n}, un subconjunto K de {1, 2, .., n} y dos enteros c y d. El certificador vertifica que la sumatoria de los elementos en las posiciones indicadas por el conjunto K del conjunto P sea menor o igual a "c" y que la sumatoria de los elementos en las posiciones indicadas por el conjunto K del conjunto V sea mayor o igual a "d". Esto se puede hacer en tiempo polinomial, ya que es solo evaluar una sumatoria.

2. Para resolver el problema SUBSET-SUM mediante una caja negra que soluciona al problema MOCHILA-VALIOSA haremos lo siguiente. Primero, hacemos que el conjunto P que recibe MOCHILA-VALIOSA sea igual al conjunto A = {a_1, a_2, ..., a_n} que recibe el problema SUBSET-SUM y hacemos que el "c" de MOCHILA-VALIOSA sea igual al "c" de SUBSET-SUM. Segundo, hacemos que el conjunto V que recibe MOCHILA-VALIOSA sea igual al conjunto A que recibe SUBSET-SUM y hacemos que el "d" de MOCHILA-VALIOSA sea igual al "c" de SUBSET-SUM. Finalmente, hacemos que el conjunto K tenga todos los índices de los elementos de A = {a_1, a_2, ..., a_n}. Luego, si MOCHILA-VALIOSA devuelve SÍ, entonces también devolvemos SÍ para SUBSET-SUM y lo mismo si la respuesta es NO.

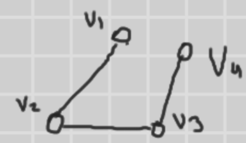
Ejercicio 6. Sabiendo que CAMINO-HAM es NP-completo, muestre que CAMINO-LARGO es NP-completo

1. CAMINO-LARGO $\in$ NP.
2. CAMINO-HAM $\leq_p$ CAMINO-LARGO.

1. Para construir el certificador B, este recibiría un certificado que consta de un grafo G, un número K y una permutación de los vértices de G. El certificador B confirma, en tiempo polinomial, que la permutación de vértices es un camino simple y tiene por lo menos K aristas.

$k=3$

$(v_1, v_2)(v_2, v_3)(v_3, v_4)$



2. Queremos solucionar CAMINO-HAM a partir de CAMINO-LARGO. Para esto, hacemos que el grafo G que recibe CAMINO-LARGO sea igual al grafo G que recibe CAMINO-HAM. Luego, hacemos que el número K que recibe CAMINO-LARGO, sea igual al número de vértices de G menos 1 (V(G)-1), ya que un camino hamiltoniano que pasa por V-1 aristas tiene V vértices, y si el número de vértices que CAMINO-LARGO tendrá es igual al número de vértices de un grafo G, entonces ese camino es un camino hamiltoniano en G.

Ejercicio 1. Sabiendo que CICLO-HAM es NP-completo, muestre que TSP es NP-completo

1. TSP $\in$ NP.
2. CICLO-HAM $\leq_p$ TSP

1. Construimos un certificador B que recibe un certificado que consta de un grafo G con pesos en las aristas, un número W y una permutación de los vértices de G. B verifica en tiempo polinomial que la permutación de los vértices represente un ciclo y que el peso total de las aristas en ese ciclo, sea menor o igual a W.

2. Queremos usar TSP para resolver el problema CICLO-HAM. Para esto, transformaremos el grafo G que recibe CICLO-HAM a un grafo completo G' donde el peso de cada arista de G' es igual a 0 si pertenece a G y 1 si no pertenece a G. Finalmente, asignamos W = 0. Si el TSP devuelve SÍ, significa que existe un ciclo que pasa por todos los vértices de G (ya que W=0 y como solo las aristas pertenecientes a G tienen peso 0, TSP devolverá un ciclo hamiltoniano en G), por lo tanto, tendríamos un ciclo hamiltoniano en G.

4 Material Adicional

- **Tardos:** Capítulo 6 y 8.
- **Cormen:** Capítulo 15, 17, 34 y 29.
- **Link al Jamboard de Reducciones:** `https://jamboard.google.com/d/1xdBeApjqZHJlByaOI-LJij-8vQsgtPh` `edit?usp=sharing`.
- **Link al Jamboard de DP 1:** `https://jamboard.google.com/d/1l7K3i7-ul6VtgC5qOgYv9hULHuWt4uKlwe4Znu` `edit?usp=sharing`.
- **Link al Jamboard de DP 2:** `https://jamboard.google.com/d/1_Ky2YpqZmKTL_6UCtVEzGYGFz2uJLT5Tkq3DP4` `s/edit?usp=sharing`.