Estrucuturas de Datos y Algoritmos II
Trabajo Práctico nº2
Cálculo de costos

Agustin Gurvich, Santiago Wirzt

12 de junio de 2020

1. Implementación de Listas

Durante toda esta sección usaremos k y h para representar el trabajo y profuncidad de (:) respectivamente.

1.1. mapS

Sean n el largo de la secuencia a ingresar (de la forma $[x_n, x_{n-1}, ..., x_1]$) y f la función a mapear, la recursión de el trabajo queda expresada como:

$$W_{mapS}(0, f) = c_0$$

$$W_{mapS}(n, f) = W_f(x_n) + W_{mapS}(n-1) + k$$

Queremos demostrar que $W_{mapS}(n, f) \in \mathcal{O}(\sum_{i=1}^{n} W_f(x_i))$.

1. (HI) Supongo que para todo $k \leq n$, $\exists c \in \mathcal{R}^+ \ / \ W_{mapS}(k, f) \leq c \cdot \sum_{i=1}^{k} W_f(x_i)$
2. (PI)

$$W_{mapS}(n+1, f) = W_f(x_{n+1}) + W_{mapS}(n, f) + k \leq^{(HI)} W_f(x_{n+1}) + c \cdot \sum_{i=1}^{n} W_f(x_i) + k \leq$$

$$\leq c' \cdot W_f(x_{n+1}) + c' \cdot \sum_{i=1}^{n} W_f(x_i) = c \cdot \sum_{i=1}^{n+1} W_f(x_i)$$

$c' \cdot W_f(x_{n+1}) \geq W_f(x_{n+1}) + k$
$c' \geq 1 + \frac{k}{W_f(x_{n+1})}$
Como $W_f(x_{n+1}) \geq 1$, me basta
$c' \geq 1 + k$

Entonces tomando $c' \geq 1 + k$ obtenemos:

$$W_{mapS}(n, f) \in \mathcal{O}(\sum_{i=1}^{n} W_f(x_i))$$

La recursión de la profundidad se expresa:

$$S_{mapS}(0, f) = c_1$$

$$S_{mapS}(n, f) = max(S_f(x_n), S_{mapS}(n-1)) + h \leq {}^1 S_f(x_n) + S_{mapS}(n-1) + h$$

Luego la recursión tiene una forma similar a W_{mapS}, entonces podemos decir:

$$S_{mapS}(n, f) \in \mathcal{O}(\sum_{i=1}^{n} S_f(x_i))$$

[1] La profundidad es siempre positiva

1.2. appendS

La implementación de listas itera sobre el primer elemento sin importar el largo del segundo. Entonces sea n el largo de la primer secuencia a ingresar:

$$W_{appendS}(0) = c_2$$

$$W_{appendS}(n) = W_{appendS}(n-1) + k$$

Esta recursión es lineal, por ende podemos decir:

$$W_{appendS}(n) \in \mathcal{O}(n)$$

La función no aplica ningún tipo de paralelización $\rightarrow S_{appendS}(n) = W_{appendS}(n)$. Por lo tanto:

$$S_{appendS}(n) \in \mathcal{O}(n)$$

1.3. reduceS

Supondremos que para la función f, $W_f(n) \in \mathcal{O}(1)$ y $S_f(n) \in \mathcal{O}(1)$. Para evaluar el costo de reduceS, primero tendremos que evaluar los de contract.

1.3.1. contract

Sea n el largo de la secuencia a ingresar. Calculamos el trabajo de contract:

$$W_{contract}(0) = c_3$$

$$W_{contract}(1) = c_4$$

$$W_{contract}(n) = W_f(x_n, x_{n-1}) + W_{contract}(n-2) + k =^2\ c_f + W_{contract}(n-2) + k$$

Queremos ver que $W_{contract}(n) \in \Theta(n)$, entonces tenemos que probar que $W_{contract}(n) \in \mathcal{O}(n)$ y $W_{contract}(n) \in \Omega(n)$. Necesitamos que esto ocurra para poder aplciar el teorema maestro más adelante.

(HI) Supongo que $\forall k \leq n\ W_{contract}(k) \in \Theta(n)$

$\mathcal{O}) W_{contract}(n+1) = c_f + W_{contract}(n-1) + k \leq^{(HI)} c_f + c \cdot (n-1) + k \leq c' + c' \cdot (n-1) + c' = c' \cdot (n+1)$
$\Omega) W_{contract}(n+1) = c_f + W_{contract}(n-1) + k \geq^{(HI)} c_f + c_m \cdot (n-1) + k \geq c'' + c'' \cdot (n-1) + c'' = c'' \cdot (n+1)$

$c' = max(c_f, c, k)$
$c'' = min(c_f, c_m, k)$

$$\therefore W_{contract}(n) \in \Theta(n)$$

Calculamos la profundidad de contract:

$$S_{contract}(0) = c_5$$

$$S_{contract}(1) = c_6$$

$$S_{contract}(n) = max(S_f(x_n, x_{n-1}), S_{contract}(n-2)) + h =^2 max(cs_f, S_{contract}(n-2)) + h$$

Queremos ver que $S_{contract}(n) \in \mathcal{O}(n)$.

(HI) Supongo que $\forall k \leq n\ S_{contract}(k) \in \mathcal{O}(n)$

$S_{contract}(n+1) = max(cs_f, S_{contract}(n-1)) + h \leq^{(HI)} max(cs_f, c \cdot (n-1)) + h =^{c > cs_f} c \cdot (n-1) + h \leq c' \cdot (n-1) + c' = c' \cdot n \leq c' \cdot (n+1)$

$c' = max(c, h)$

$$\therefore S_{contract}(n) \in \mathcal{O}(n)$$

[2] f es constante

1.3.2. Costo

Procedemos a calcular los costos de reduceS. Sea n el largo de la secuencia a ingresar:

$$W_{reduceS}(0) = c_7$$

$$W_{reduceS}(1) = W_f(x) =^3 \ c_8$$

$$W_{reduceS}(n) = W_{contract}(n) + W_{reduceS}(\lceil n/2 \rceil) + k$$

Suponiendo $n = 2^p$ (H)

$$W_{reduceS}(n) = W_{contract}(n) + W_{reduceS}(\lceil n/2 \rceil) + k =^{(H)} \ W_{contract}(n) + W_{reduceS}(n/2) + k$$

Tomando $a = 1, b = 2, \epsilon = 1, N > 1$ y $c = \frac{1}{2}$, luego podemos aplicar la tercer regla del teorema maestro. Y tenemos:

$$W_{contract}(n) \in \Omega(n^{(\log_2 1)+1}) = \Omega(n)$$

$$W_{contract}(\frac{n}{2}) =^4 \ \frac{n}{2} = \frac{1}{2} \cdot n = \frac{1}{2} W_{contract}(n)$$

$$\therefore W_{reduceS}(2^p) \in \Theta(W_{contract}(2^p)) = \Theta(2^p)$$

Como la función identidad es suave y $W_{reduceS}$ es no decreciente por ser una función de costos, podemos expandir la definición:

$$\therefore \forall n \, W_{reduceS}(n) \in \Theta(n)$$

La función no aplica ningún tipo de paralelización $\rightarrow S_{reduceS}(n) = W_{reduceS}(n)$. Por lo tanto:

$$\therefore \forall n \, S_{reduceS}(n) \in \Theta(n)$$

1.4. scanS

Supondremos que para la función f, $W_f(n) \in \mathcal{O}(1)$ y $S_f(n) \in \mathcal{O}(1)$. scanS hace uso de las funciones contract y expand, pero ya analizamos contract en 1.3.1. Procedemos a calcular los costos de expand.

1.4.1. expand

La función de expand itera sobre los elementos de la primer secuencia ingresada, ya que espera que la segunda secuencia tenga los elementos necesarios para el cálculo. Entonces sea n el largo de la secuencia:

$$W_{expand}(0) = c_9$$

$$W_{expand}(1) = c_{10}$$

$$W_{expand}(n) = W_f(x_n, y) + 2 \cdot k + W_{expand}(n-2) =^5 \ c_f + c \cdot k + W_{expand}(n-2)$$

Queremos ver que $W_{expand}(n) \in \Theta(n)$, entonces tenemos que probar que $W_{expand}(n) \in \mathcal{O}(n)$ y $W_{expand}(n) \in \Omega(n)$. Necesitamos que esto ocurra para poder aplciar el teorema maestro más adelante.

(HI) Supongo que $\forall k \leq n \, W_{expand}(k) \in \Theta(n)$

$\mathcal{O}) W_{expand}(n+1) = c_f + W_{expand}(n-1) + 2 \cdot k \leq^{(HI)} \ c_f + c \cdot (n-1) + 2 \cdot k \leq c' + c' \cdot (n-1) + c' = c' \cdot (n+1)$
$\Omega) W_{expand}(n+1) = c_f + W_{expand}(n-1) + 2 \cdot k \geq^{(HI)} \ c_f + c_m \cdot (n-1) + 2 \cdot k \geq c'' + c'' \cdot (n-1) + c'' = c'' \cdot (n+1)$

$$c' = max(c_f, c, 2 \cdot k)$$
$$c'' = min(c_f, c_m, 2 \cdot k)$$

[3] f es constante
[4] trabajo de contract
[5] f es constante

$$\therefore W_{expand}(n) \in \Theta(n)$$

.

$$S_{expand}(0) = c_{11}$$
$$S_{expand}(1) = c_{12}$$
$$S_{expand}(n) = max(S_f(x_n, x_{n-1}), S_{expand}(n-2)) + 2 \cdot h =^{6} max(cs_f, S_{expand}(n-2)) + 2 \cdot h$$

Queremos ver que $S_{expand}(n) \in \mathcal{O}(n)$.

(HI) Supongo que $\forall k \leq n \; S_{expand}(k) \in \mathcal{O}(n)$

$S_{expand}(n+1) = max(cs_f, S_{expand}(n-1)) + 2 \cdot h \leq^{(HI)} max(cs_f, c \cdot (n-1)) + 2 \cdot h =^{c > cs_f} c \cdot (n-1) + 2 \cdot h \leq c' \cdot (n-1) + c' = c' \cdot n \leq c' \cdot (n+1)$

$$c' = max(c, 2 \cdot h)$$

$$\therefore S_{expand}(n) \in \mathcal{O}(n)$$

1.4.2. Costo

Sea n el largo de la sequencia a evaluar:

$$W_{scanS}(0) = c_{13}$$
$$W_{scanS}(0) = c_{14}$$

$$W_{scanS}(n) = W_{contract}(n) + W_{scanS}(\lceil n/2 \rceil) + W_{expand}(n) + c_{15} =^{7} W_{contract}(n) + W_{scanS}(n/2) + W_{expand}(n) + c_{15}$$

Tomando $a = 1, b = 2, \epsilon = 1, N > 1$ y $c = \frac{1}{2}$, luego podemos aplicar la tercer regla del teorema maestro.
Y tenemos:

$$W_{contract}(n) + W_{expand}(n) \in \Omega(n^{(\log_2 1)+1}) = \Omega(n)$$
$$W_{contract}(\frac{n}{2}) + W_{expand}(\frac{n}{2}) =^{8} \frac{n}{2} + \frac{n}{2} = \frac{1}{2} \cdot (n + n) = \frac{1}{2}(W_{contract}(n) + W_{expand}(n))$$
$$\therefore W_{scanS}(n) \in \Theta(W_{contract}(n) + W_{expand}(n)) = \Theta(n)$$

La función no aplica ningún tipo de paralelización $\rightarrow S_{scanS}(n) = W_{scanS}(n)$. Por lo tanto:

$$S_{scanS}(n) \in \Theta(n)$$

[6] f es constante
[7] Por regla de suavidad
[8] trabajo de contract y expand

2. Implementación de Arrays

2.1. mapS

Sea n el largo de la secuencia a ingresar (x) y f la función a mapear. Primero veremos el trabajo de la función:

$$W_{mapS}(n,f) = W_{lengthS}(n) + W_{tabulateS}(f',n) + c_0 =^{9} \mathcal{O}(1) + \mathcal{O}(\sum_{i=1}^{n} W_{f'}(x_i)) + c_0$$

Donde definimos $f' = (\backslash i \to f(nthS\, xs\, i))$. Podemos plantear su trabajo como:

$$W_{f'}(f,i) = W_f(x_i) + W_{nthS}(i) + c_0 =^{10} W_f(x_i) + \mathcal{O}(1) + c_0 =^{11} \mathcal{O}(max(W_f(x_i),1)) + c_0 = W_f(x_i)$$

pues el mínimo trabajo que puede realizar la función f es uno constante. Así $W_{f'}(f,i) \in \mathcal{O}(W_f(x_i))$
Finalmente obtenemos:

$$\mathcal{O}(1) + \mathcal{O}(\sum_{i=1}^{n} W_f(x_i)) + c_0 =^{10} max(\mathcal{O}(1), \mathcal{O}(\sum_{i=1}^{n} W_f(x_i))) + c_0 = \mathcal{O}(\sum_{i=1}^{n} W_f(x_i))$$

En segundo lugar planteamos la profundidad de la función:

$$S_{mapS}(n,f) = S_{lengthS}(n) + S_{tabulateS}(f',n) + c_0 =^{12} \mathcal{O}(1) + \mathcal{O}(max_{i=1}^{n} S_{f'}(x_i)) + c_0$$

Donde f' es la función definida previamente. Podemos plantear su profundidad como:

$$S_{f'}(f,i) = S_f(x_i) + S_{nthS}(i) + c_0 =^{13} S_f(x_i) + \mathcal{O}(1) + c_0 =^{10} \mathcal{O}(max(S_f(x_i),1)) + c_0 = S_f(x_i)$$

pues el mínimo trabajo que puede realizar la función f es uno constante. Así $S_{f'}(f,i) \in \mathcal{O}(S_f(x_i))$
Finalmente obtenemos:

$$\mathcal{O}(1) + \mathcal{O}(max_{i=1}^{n} S_f(x_i)) + c_0 =^{10} \mathcal{O}(max(1, max_{i=1}^{n} S_f(x_i))) + c_0 = \mathcal{O}(max_{i=1}^{n} S_f(x_i))$$

$$\therefore W_{mapS}(n,f) \in \mathcal{O}(W_f(x_i)) \text{ y } S_{mapS}(n,f) \in (max_{i=1}^{n} S_f(x_i))$$

2.2. appendS

Sean s, t el tamaño de las secuencias a unir. Primero veamos el trabajo de la función

$$W_{appendS}(s,t) = 2 \cdot W_{lengthS} + W_{tabulateS}(f', s+t) + c_0 =^{8} 2 * \mathcal{O}(1) + \mathcal{O}(\sum_{i=1}^{s+t} W_{f'}(x_i)) + c_0$$

Donde definimos f' = (\ i → if i < n then nthS xs i else nthS ys (i-n)) y su trabajo es:

$$W_{f'} = W_{nthS}(i) + c_1 =^{9} \mathcal{O}(1) + c_1 = \mathcal{O}(1)$$

Así podemos continuar el cálculo anterior:

$$2 * \mathcal{O}(1) + \in \mathcal{O}(\sum_{i=1}^{s+t} W_{f'}(x_i)) + c_0 = 2 * \mathcal{O}(1) + \mathcal{O}(\sum_{i=1}^{s+t} \mathcal{O}(1)) + c_0 =^{14} 2 * \mathcal{O}(1) + \mathcal{O}(s+t) + c_0 =^{15}$$

$$2 * \mathcal{O}(1) + \mathcal{O}(s+t) =^{10} \mathcal{O}(max(1, s+t)) = \mathcal{O}(s+t)$$

[9] El trabajo de lengthS y de tabulateS están dados en la tabla provista por la cátedra
[10] El trabajo de nthS está dado en la tabla provista por la cátedra
[11] Por definición de suma de complejidades
[12] La profundidad de lengthS y de tabulateS están dadas en la tabla provista por la cátedra
[13] La profundidad de nthS está dada en la tabla provista por la cátedra
[14] La sumatoria está acotada por (s+t)
[15] Podemos obviar la constante c

Ahora veamos la profundidad de la función:

$$S_{appendS}(s,t) = 2 \cdot S_{lengthS} + S_{tabulateS}(f', s+t) + c_0 =^{11} 2 * \mathcal{O}(1) + \mathcal{O}(\max_{i=1}^{s+t} S_{f'}(x_i)) + c_0$$

Donde f' es la función definida anteriormente y su profundidad es:

$$S_{f'}(i) = S_{nthS}(i) + c_1 =^{16} \mathcal{O}(1) + c_1 = \mathcal{O}(1)$$

Así, continuando el cálculo anterior:

$$2 * \mathcal{O}(1) + \mathcal{O}(\max_{i=1}^{s+t-1} S_{f'}(x_i)) + c_0 = 2 * \mathcal{O}(1) + \mathcal{O}(\max_{i=1}^{s+t-1} 1) + c_0 =^{14} 2 * \mathcal{O}(1) + \mathcal{O}(1) = 3 * \mathcal{O}(1) = \mathcal{O}(1)$$

$$\therefore W_{appendS}(s+t) \in \mathcal{O}(s+t) \text{ y } S_{appendS}(s,t) \in \mathcal{O}(1)$$

2.3. reduceS

Sean n el tamaño de la secuencia a reducir y f la función a aplicar. Debemos suponer que $f \in \mathcal{O}(1)$. Para poder analizar correctamente su costo, primero debemos analizar la función auxiliar *contract*:

$$W_{contract}(f,n) = W_{lengthS}(n) + W_{tabulateS}(f', n) + c_0 =^{17} \Theta(1) + \Theta(\sum_{i=1}^{n/2+1} W_{f'}(x_i)) + c_0$$

Según la paridad del argumento n, f' puede tomar dos formas distintas

- Si n es par, entonces f' = (\ i → f (nthS xs (2*i)) (nthS xs (2*i+1)))
- Si n es impar, entonces f' = (\ i → if i == n then nthS xs (2*i) else f (nthS xs (2*i)) (nthS xs (2*i+1)))

Como el trabajo de f y el trabajo de $nthS$ son constantes, podemos decir que el trabajo de f' es constante sin importar su forma. Es decir, $W_{f'} \in \Theta(1)$
Continuando con el cálculo anterior:

$$\Theta(1)+\Theta(\sum_{i=1}^{n/2+1} W_{f'}(x_i))+c_0 = \Theta(1)+\Theta(\sum_{i=1}^{n/2+1} 1)+c_0 = \Theta(1)+\Theta(n/2+1)+c_0 =^{19} \Theta(1)+\Theta(n/2+1) =^{18}$$

$$\Theta(max(1, n/2+1)) = \Theta(n/2+1)$$

Pero también sabemos que $n/2+1 \in \Theta(n)$. Luego $W_{contract}(f,n) \in \Theta(n)$
Veamos ahora la profundidad de contract:

$$S_{contract}(f,n) = S_{lengthS}(n) + S_{tabulateS}(f', n) + c_0 =^{19} \Theta(1) + \Theta(max_{i=1}^{n/2+1} S_{f'}(x_i)) + c_0$$

Al igual que en el caso anterior, f' va a tomar las dos formas distintas según la paridad de n. Como la profundidad de f y la profundidad de $nthS$ son constantes, podemos decir que la profundidad de f' es constante sin importar su forma. Es decir, $S'_f \in \Theta(1)$
Retomando el cálculo anterior:

$$\Theta(1)+\Theta(max_{i=1}^{n/2+1} S_{f'}(x_i))+c_0 = \Theta(1)+\Theta(max_{i=1}^{n/2+1} 1)+c_0 = \Theta(1)+\Theta(1)+c_0 =^{20} \Theta(1)+\Theta(1) = 2*\Theta(1)$$

Como sabemos que $2 * \Theta(1) \in \Theta(1)$, podemos concluir que $S_{contract}(f,n) \in \Theta(1)$

$$\therefore W_{contract}(f,n) \in \mathcal{O}(n) \text{ y } S_{contract}(f,n) \in \mathcal{O}(1)$$

Ahora estamos en condiciones de analizar el costo de reduceS. Primero veamos su trabajo, por inducción sobre el largo de la secuencia:

[16] La profundidad de nthS está dada en la tabla provista por la cátedra
[17] El trabajo de lengthS y tabulateS están dados en la tabla provista por la cátedra
[18] Por definición de la suma de complejidades
[19] El trabajo de lengthS y tabulateS están dados en la tabla provista por la cátedra
[20] Podemos obviar la constante c

- $W_{reduceS}(f, 0) = c_0$
- $W_{reduceS}(f, 1) = W_f(x_i) + W_{nthS}(1) + c_1$
- $W_{reduceS}(f, n) = W_{contract}(f, n) + W_{reduceS}(f, \lceil n/2 \rceil) + c_2$

Para utilizar el teorema de suavidad, supongamos que $n = 2^k$. Con esto podemos reescribir los costos como:

- $W_{reduceS}(f, 0) = c_0$
- $W_{reduceS}(f, 1) = W_f(x_i) + W_{nthS}(1) + c_1$
- $W_{reduceS}(f, n) = W_{contract}(f, n) + W_{reduceS}(f, n/2) + c_2$

Utilizando la tabla de costos provista por la cátedra podemos ya resolver ciertos costos:

- $W_{reduceS}(f, 0) = c_0$
- $W_{reduceS}(f, 1) = \mathcal{O}(1) + \mathcal{O}(1) + c_1 = \mathcal{O}(1)$

Para el caso donde $n > 1$, sabemos que $W_{contract}(f, n) \in \Theta(n)$. Identificando $a = 1, b = 2$ podemos aplicar el teorema maestro. Queremos aplicar el tercer caso del teorema, así que debemos buscar un $\epsilon > 0$ tal que $W_{contract}(f, n) \in \Omega(n^{log_2(1)+\epsilon})$ y un $c < 1, N \in \mathbb{N}$ tal que $\forall n > N, contract(f, n/2) \leq c * contract(f, n)$.

- Como $n^{log_2(1)} = n^0$, tomando $\epsilon = 1$ se tiene que $n^{log_2(1)+1} = n$ y como $W_{contract}(f, n) \in \Theta(n)$, entonces $W_{contract}(f, n) \in \Omega(n)$
- Sabemos que el valor de $W_{contract}(f, n)$ aumenta a medida que aumenta el tamaño de la secuencia de entrada. Así, si recibe la mitad de la entrada, tardará la mitad del tiempo (pues su trabajo es lineal). Basta tomar $c = 1/2, N = 1$ para cumplir la desigualdad

Por lo tanto podemos decir, por el teorema maestro, que $W_{reduceS}(f, 2^k) \in \mathcal{O}(n)$
Faltaría ver que $W_{reduceS}(f, n)$ es no decreciente para completar el teorema de suavidad. Esto es más directo, ya que a medida que aumenta el tamaño de la secuencia también aumentan la cantidad de operaciones a realizar.
Como $\mathcal{O}(1) \subseteq \mathcal{O}(n)$, el caso de la secuencia de largo 1 está contemplado en la complejidad. Luego podemos afirmar sin inconvenientes que $W_{reduceS}(f, n) \in \mathcal{O}(n)$
Queremos ahora analizar la profundidad de reduceS, y lo veremos por inducción sobre el largo de la secuencia:

- $S_{reduceS}(f, 0) = c_0$
- $S_{reduceS}(f, 1) = S_f(x_i) + S_{nthS}(1) + c_1$
- $S_{reduceS}(f, n) = S_{contract}(f, n) + S_{reduceS}(f, \lceil n/2 \rceil) + c_2$

Para utilizar el teorema de suavidad, supongamos que $n = 2^k$. Con esto podemos reescribir los costos como:

- $S_{reduceS}(f, 0) = c_0$
- $S_{reduceS}(f, 1) = S_f(x_i) + S_{nthS}(1) + c_1$
- $S_{reduceS}(f, n) = S_{contract}(f, n) + S_{reduceS}(f, n/2) + c_2$

Utilizando la tabla de costos provista por la cátedra podemos ya resolver ciertos costos:

- $S_{reduceS}(f, 0) = c_0$
- $S_{reduceS}(f, 1) = \mathcal{O}(1) + \mathcal{O}(1) + c_1 = \mathcal{O}(1)$

Para el caso donde $n > 1$, sabemos que $S_{contract}(f,n) \in \mathcal{O}(1)$. Identificando $a = 1, b = 2$ podemos aplicar el teorema maestro. Queremos utilizar el segundo caso, por lo que basta ver que $S_{contract}(f,n) \in \mathcal{O}(n^{log_2(1)})$. Como $log_2(1) = 0$, debería ser que $S_{contract}(f,n) \in \mathcal{O}(1)$ y eso fue demostrado anteriormente. Por lo tanto $S_{reduceS}(f, 2^k) \in \mathcal{O}(lg(n))$.
Faltaría ver que $S_{reduceS}(f,n)$ es no decreciente para completar el teorema de suavidad. Esto es más directo, ya que al igual que en el caso del trabajo, la función no decrementa cuando aumenta el largo de la secuecnia de entrada.
Como $\mathcal{O}(1) \subseteq \mathcal{O}(lg(n))$, el caso de la secuencia de largo 1 está contemplado en la complejidad.
Luego podemos afirmar sin inconvenientes que $S_{reduceS}(f,n) \in \mathcal{O}(lg(n))$

$$\therefore W_{reduceS}(f,n) \in \mathcal{O}(n) \text{ y } S_{reduceS}(f,n) \in \mathcal{O}(lg(n))$$

2.4. scanS

Sean n el tamaño de la secuencia a operar y f la función a aplicar. Debemos suponer que $f \in \mathcal{O}(1)$. En el apartado anterior ya analizamos los costos de *contract*, así que nos falta analizar *expand*. Resaltamos que *expand* depende de la longitud de la primer secuencia argumento:

$$W_{expand}(f,n) = W_{lengthS}(n) + W_{tabulateS}(combinArr, n) + c_0 = {}^{21}\Theta(1) + \Theta(\sum_{i=1}^{n} W_{combinArr}(x_i)) + c_0$$

Donde definimos *combinArr* como:

```
combinArr i = if even i then (nthS ys (div i 2))
                        else f (nthS ys (div i 2)) (nthS xs (i-1))
```

Observamos que el trabajo de todas las funciones involucradas es constante, por lo que $W_{combinArr} \in \Theta(1)$
Retomando el cálculo anterior:

$$\Theta(1)+\Theta(\sum_{i=1}^{n} W_{combinArr}(x_i))+c_0 = \Theta(1)+\Theta(\sum_{i=1}^{n} \Theta(1))+c_0 = \Theta(1)+\Theta(n)+c_0 = {}^{22}\Theta(1)+\Theta(n) = {}^{23}\Theta(max(1,n)) = \Theta(n)$$

Por lo tanto $W_{expand}(f,n) \in \Theta(n)$
Veamos ahora la profundidad de expand:

$$S_{expand}(f,n) = S_{lengthS}(n) + S_{tabulateS}(combinArr, n) + c_0 = {}^{24}\Theta(1) + \Theta(max_{i=1}^{n} S_{combinArr}(x_i)) + c_0$$

Al igual que en el cálculo del trabajo, las profundidades involucradas en la función *combinArr* son constantes. Por lo tanto $S_{combinArr(f,n)} \in \Theta(1)$. Continuando con el cálculo anterior:

$$\Theta(1) + \Theta(max_{i=1}^{n} S_{combinArr}(x_i)) + c_0 = \Theta(1) + \Theta(max_{i=1}^{n} 1) + c_0 = \Theta(1) + \Theta(1) + c_0 = {}^{21} = 2 * \Theta(1)$$

Como $2 * \Theta(1) \in \Theta(1)$, tenemos que $S_{expand}(f,n) \in \Theta(1)$

$$\therefore W_{expand}(f,n) \in \mathcal{O}(n) y S_{expand}(f,n) \in \mathcal{O}(1)$$

Estamos ahora en condiciones de calcular los costos de *scanS*. Veamos primero el trabajo. Veamos cómo se comporta según el largo de la secuencia:

- $W_{scanS}(f,0) = W_{singletonS}(1) + c_0$
- $W_{scanS}(f,1) = W_{singletonS}(1) + W_f(x_i) + W_{nthS}(0)$
- $W_{scanS}(f,n) = W_{contract}(f,n) + W_{scanS}(f, \lceil n/2 \rceil) + W_{expand}(f,n) + c_2$

Para utilizar el teorema de suavidad, supongamos que $n = 2^k$. Con esto podemos reescribir los costos como:

[21] El trabajo de lengthS y de tabulateS están dados en la tabla provista por la cátedra
[22] Podemos obviar la constante c
[23] por definicion de la suma de complejidades
[24] La profundidad de lengthS y de tabulateS están dados en la tabla provista por la cátedra

- $W_{scanS}(f,0) = W_{singletonS}(1) + c_0$
- $W_{scanS}(f,1) = W_{singletonS}(1) + W_f(x_i) + W_{nthS}(0) + c_1$
- $W_{scanS}(f,n) = W_{contract}(f,n) + W_{scanS}(f,n/2) + W_{expand}(f,n) + c_2$

Con la tabla provista por la cátedra podemos ya resolver ciertos costos:

- $W_{scanS}(f,0) = \mathcal{O}(1) + c_0$
- $W_{scanS}(f,1) = \mathcal{O}(1) + \mathcal{O}(1) + \mathcal{O}(1) + c_1$

Y resultan

- $W_{scanS}(f,0) \in \mathcal{O}(1)$
- $W_{scanS}(f,1) \in \mathcal{O}(1)$

Para el trabajo si $n > 1$, ya sabemos que los trabajos de *contract* y *expand* son lineales, por lo que su suma también será lineal. Ahora podemos aplicar el tercer caso del teorema maestro. Identificando $a = 1$ y $b = 2$ debemos buscar un $\epsilon > 0$ tal que $W_{contract}(f,n) + W_{expand}(f,n) \in \Omega(n^{log_2(1)+\epsilon})$ y un $c < 1, N \in \mathbb{N}$ tal que $\forall n > N, W_{contract}(f,n/2) + W_{expand}(f,n/2) \leq c * (W_{contract}(f,n) + W_{expand}(f,n))$.

- Tanto $W_{contract}(f,n)$ como $W_{expand}(f,n)$ pertenecen a $\Theta(n)$, en particular ambas son $\Omega(n)$. Luego $W_{contract}(f,n) + W_{expand}(f,n) \in \Omega(n)$ por lo que $\epsilon = 1$
- Sabemos que los valores de $W_{contract}(f,n)$ y $W_{expand}(f,n)$ aumentan a medida que aumenta el tamaño de la secuencia de entrada. Así, si se recibe la mitad de la entrada, tardará la mitad del tiempo (pues su trabajo es lineal). Basta tomar $c = 1/2, N = 1$ para cumplir la desigualdad.

Así podemos decir por el teorema maestro que $W_{scanS}(f, 2^k) \in \mathcal{O}(n)$.
Faltaría ver que es no decreciente, pero esto es más directo ya que a medida que aumenta el tamaño de la secuencia también aumenta la cantidad de operaciones a realizar.
Además, como $\mathcal{O}(1) \subseteq \mathcal{O}(n)$, el caso de una secuencia de longitud menor o igual a 1 también está considerado en la complejidad.
Finalmente podemos afirmar, por el teorema de suavidad, que $W_{scanS}(f,n) \in \mathcal{O}(n)$
Queremos ahora analizar la profundidad de *scanS*, y lo veremos por inducción sobre el largo de la secuencia:

- $S_{scanS}(f,0) = S_{singletonS}(1) + c_0$
- $S_{scanS}(f,1) = S_{singletonS}(1) + S_f + S_{nthS}(0)$
- $S_{scanS}(f,n) = S_{contract}(f,n) + S_{scanS}(f, \lceil n/2 \rceil) + S_{expand}(f,n) + c_2$

Para utilizar el teorema de suavidad, supongamos que $n = 2^k$. Con esto podemos reescribir los costos como:

- $S_{scanS}(f,0) = S_{singletonS}(1) + c_0$
- $S_{scanS}(f,1) = S_{singletonS}(1) + S_f + S_{nthS}(0) + c_1$
- $S_{scanS}(f,n) = S_{contract}(f,n) + S_{scanS}(f,n/2) + S_{expand}(f,n) + c_2$

Con la tabla provista por la cátedra podemos ya resolver ciertos costos:

- $S_{scanS}(f,0) = \mathcal{O}(1) + c_0$
- $S_{scanS}(f,1) = \mathcal{O}(1) + \mathcal{O}(1) + \mathcal{O}(1) + c_1$

Para el caso donde $n > 1$, sabemos que $S_{contract}(f,n) \in \mathcal{O}(1)$ y $S_{expand}(f,n) \in \mathcal{O}(1)$. Identificando $a = 1, b = 2$ podemos aplicar el teorema maestro. Queremos utilizar el segundo caso, por lo que basta ver que $S_{contract}(f,n) + S_{expand}(f,n) \in \mathcal{O}(n^{log_2(1)})$. Como $log_2(1) = 0$, debería ser que $S_{contract}(f,n) + S_{expand}(f,n) \in \mathcal{O}(1)$. Como ambas profundidades son constantes, su suma también lo será. Por lo tanto $S_{reduceS}(f, 2^k) \in \mathcal{O}(lg(n))$.
Faltaría ver que $S_{scanS}(f,n)$ es no decreciente para completar el teorema de suavidad. Esto es más directo, ya que al igual que en el caso del trabajo, la función no decrementa cuando aumenta el largo de

la secuencia de entrada.
Como $\mathcal{O}(1) \subseteq \mathcal{O}(lg(n))$, el caso de la secuencia de largo 1 está contemplado en la complejidad.
Luego podemos afirmar sin inconvenientes que $S_{scanS}(f, n) \in \mathcal{O}(lg(n))$

$$\therefore W_{scanS}(f, n) \in \mathcal{O}(n) \text{ y } S_{scanS}(f, n) \in \mathcal{O}(lg(n))$$