Reporte 2: Algoritmos de Ordenamiento

Ricardo Montiel Manriquez

29 de Julio del 2021

1 Descripcion del Algoritmo Shell Sort

Es un algoritmo de ordenamiento y se le nombro Shell Sort en honor a su inventor Donald Shell. Su implementación original, requiere O(n2) comparaciones e intercambios en el peor caso, aunque un cambio menor presentado en el libro de V. Pratt produce una implementación con un rendimiento de $O(nlog^2n)$ en el peor caso. Este algoritmo es una generalización del ordenamiento por inserción y también se le conoce como inserción con incrementos decrecientes, teniendo en cuenta dos observaciones:

- El ordenamiento por inserción es eficiente si la entrada está "casi ordenada".
- El ordenamiento por inserción es ineficiente, en general, porque mueve los valores sólo una posición cada vez.

Shell Sort es una mejora de la ordenación por inserción, en donde se van comparando elementos distantes, al tiempo que se los intercambian si corresponde. A medida que se aumentan los pasos, el tamaño de los saltos disminuye, por esto mismo, es útil tanto como si los datos desordenados se encuentran cercanos, o lejanos.

Por ejemplo:

Original	81	94	11	96	12	35	17	95	28	58	41	75	15
After 5-sort	35	17	11	28	12	41	75	15	96	58	81	94	95
After 3-sort	28	12	11	35	15	41	58	17	94	75	81	96	95
After 1-sort	11	12	15	17	28	35	41	58	75	81	94	95	96

En esta tabla que vemos en el libro ShellSort-Weiss-3rd-edition se nos da un ejemplo de su funcionamiento, donde primero empieza con distancias de 5, luego con distancias de 3 y finalmente con distancias de 1. Finalmente ordenando el arreglo. Hay que aclarar que esta no es la secuencia que sugiere Shell, pero sirve para dar una idea de su funcionamiento.

2 La aplicación del algoritmo sobre un ejemplar distinto

Tengamos el siguiente arreglo: A = {5, 8, 2, 1, 6, 7, 9, 3, 4, 3} y siguiendo la secuencia sugeria por Shell $h_t = N/2$, *and* $[h_k = h_{k+1}/2]$, obtenemos lo siguiente:

Posision del Arreglo	0	1	2	3	4	5	6	7	8	9
Arreglo a Ordenar	5	8	2	1	6	7	9	3	4	3
k = 10 ⇒ k = 10/2 = 5	5	8	2	1	3	7	9	3	4	6
k = 5 ⇒ k = 5/2 = 2.5	2	1	3	3	4	6	5	7	9	8
k = 2 ⇒ k = 2/2 = 1	1	2	3	3	4	5	6	7	8	9

Donde en la primera parte se hacen comparaciones en intervalos de 5, es decir, de la posicion 0 con la 5, luego la 1 con la 6, la 2 con la 7 y asi sucesivamente, y en caso de ser menor la comparacion, esta hace un intercambio con los elementos, un vez que termina de hacer las comparaciones se repite el proceso otra vez hasta que al moemnto de hacer todas la comparacines no se haga ningun intercambio.

Ahora las comparaciones se hacen en intervalos de 2, es decir de 0 con 2, 1 con 3, 2 con 4 y asi sucesivamente, se repite lo mismo anteriormente mencionado, en caso de ser menor la comparacion, esta hace un intercambio con los elementos, un vez que termina de hacer las comparaciones se repite el proceso otra vez hasta que al moemnto de hacer todas la comparacines no se haga ningun intercambio..

Y por ultimo hacemos comparaciones en intervalos de 1, es decir, de 0 con 1, de 1 con 2, 2 con 3 y asi sucesivamente, repitiendo lo anterior mencionado.

3 Análisis de complejidad en el peor caso con Theta

Nos dice que su complejidad es $\Theta(N^2)$ en el peor de los casos y la forma de demostrarlo es creando un caso en concreto donde la entrada se mala y tambien el tamaño del arreglo tiene que ser potencia de 2 para que todos los incrementos que se realicen sean forsozamente tengan que ser pares exepto el ultimo donde es 1.

Un arreglo que se da como ejemplo es uno que va de 1 hasta 16 elementos, y la forma en la que esta acomodado el arreglo es: en las posiciones del arreglo que sean pares es decir (0, 2, 4, 6, 8, 10, 12, 14, 16) iran los elementos mas grandes, mientras que en el las posiciones impares (1, 3, 5, 7, 9, 11, 13, 15) iran los elemntos mas pequeños del arreglo.

Entonces al hacer los incrementos, los elementos que se encuentren en las posiones pares siempre se compararan con los pares y lo mismo pasa con los elementos de las posiciones impares, siempre se compararan entre elementos de las posiciones impares, por lo que al momento de hacer el ultimo incremento los $N/2$ elementos mas grandes se siguen encontrando en las posisiones pares y lo mismo pasa con los $N/2$ elementos mas pequeños, estos se siguen encontrando en las posiciones impares, entonces el i-esimo elemento mas pequeño ($i <= N/2$ se encuentra en la posicion $2i-1$ y al momento de tener que ordernarlo en el lugar donde tiene que ir se requiere moverlo $i-1$ espacios que hay en el arreglo.

Por lo que al tener que acomodar a los elementos mas pequeños en el lugar correspondiente del arreglo esto toma $\sum_{N/2}^{i=1} i - 1 = \Omega\ (N^2)$.