

Ejercicio Formativo 4

(Fecha de entrega: [2021-04-21 Tue])

Sofía Valentina Bobadilla Ponce

1 Parte a

Para la solución del problema se considera la notación del modelo de memoria secundaria donde:

- M = tamaño memoria principal
- N = tamaño input
- B = tamaño de los bloques
- $m = \frac{M}{B}$
- $n = \frac{N}{B}$
- $N \gg M \gg B$

Por lo estudiado es conocido que al estar en un entorno de memoria secundaria y detener la recursión cuando los subarreglos son de tamaño B el orden de esa variante es $O(n \log(n))$.

El método external merge sort lleva la información del input a la memoria en forma de bloques y los compara, el método general realiza esto de a 2 bloques, sin embargo, para minimizar los costos de acceso a la memoria tiene sentido aumentar el tamaño de los bloques, si se aplica la variante de detener la recursión cuando los bloques son de tamaño M la cantidad de I/Os se reduce a $O(n \log(\frac{n}{m}))$.

Ahora otra forma de optimizar los accesos es considerar la cantidad de bloques que se llevan a memoria, sería óptimo tener mas de 2 buffer de lectura, tal que cada uno lleva un bloque a memoria (estos estarían en la memoria secundaria) se comparan y se va escribiendo sobre el archivo de salida. Es importante considerar que la idea anterior es limitada ya que la cantidad de buffers que se podrían añadir no pueden ser mayores a $\frac{M}{B}$, es decir, se encontró una cantidad máxima de buffers. Con este cambio el orden pasa a ser $O(n \log_m(\frac{n}{m}))$ el cual se simplifica a $O(n \log_m(n))$.

Lo anterior fue estudiado en clases y visto en el apunte, sigue ahora calcular el costo para las condiciones del ejercicio. Se tiene la variable ρ que implicará la cantidad de subarreglos ordenados dentro del input, en primer lugar el costo de calcular esto será de orden $O(n)$, luego de eso se define que se detendrá la recursión cuando el tamaño de los bloques sea $\frac{N}{\rho}$ lo cual se podría traducir a que se espera que al fraccionar el input por la cantidad de subarreglos ordenados los costos de comparación y escritura se vean reducidos, con ello se tendría que la cantidad de I/Os se reduce a $O(n \log(\frac{N}{B})) = O(n \log(\frac{N}{\rho})) = O(n \log(\rho))$, luego se maximiza la cantidad de bloques bajo los mismo términos explicados previamente lo que le añade un orden al logaritmo y nos entrega el orden buscado considerando los costos del cálculo de ρ y el de ordenamiento en memoria externa: $O(n + n \log_m(\rho)) = O(n(1 + \log_m(\rho)))$

2 Parte b

En clases se han visto los metodos de reducción y teoría de la información para la búsqueda de cotas inferiores, para demostrar lo que se pide en el ejercicio basta con encontrar una cota inferior para el problema mayor a $O(n)$.

Lo visto en la parte a entrega el orden para ordenar A el cual considera el orden de calcular ρ y el ordenamiento en la memoria externa, en particular el segundo orden corresponde al cálculo de una cota inferior, es decir, al menos se tendrá ese costo para ese proceso, con esto sin importar lo que diga el amigo no se puede resolver el problema con $O(n)$.