Universidad Centroamericana José Simeón Cañas
Departamento de Electrónica e Informática
Facultad de Ingeniería y Arquitectura
Análisis de Algoritmos **Guía de Ejercicios No. 5**
Ciclo 02/2024 **Bitmasks, Fenwick Trees y Memoization**

Bitmasks

1. Comprobar si S es par usando operaciones con bits.
2. Calcular el número de 1s (bits activados) en la representación binaria de S .
3. Encontrar el bit más significativo (MSB) activado en S .
4. Encontrar el bit menos significativo (LSB) activado en S .
5. Apagar (desactivar) el LSB en S .
6. Calcular el XOR de todos los números desde 1 hasta N de manera eficiente.
7. Intercambiar dos variables sin usar una variable temporal.
8. Verificar si dos enteros tienen signos opuestos.
9. Activar la última secuencia consecutiva de 1s en S .
10. Activar la última secuencia consecutiva de 0s en S .
11. Extraer los últimos K bits de S .
12. Multiplicar S por 2^N usando un desplazamiento a la izquierda.
13. Dividir S por 2^N usando un desplazamiento a la derecha.
14. Crear una máscara con solo los primeros K bits activados.
15. Crear una máscara con solo los últimos K bits activados.
16. Aplicar una máscara para apagar todos los bits excepto los primeros K en S .
17. Aplicar una máscara para apagar todos los bits excepto los últimos K en S .
18. Crear una máscara para activar todos los bits desde P1 hasta P2 .
19. Desplazar los bits de S a la izquierda N posiciones.
20. Desplazar los bits de S a la derecha N posiciones.
21. Determinar si S es un subconjunto de T.
22. Contar cuántos bits difieren entre S y T.
23. Encontrar la primera posición donde S y T difieren.
24. Verificar si todos los bits activados en T también están activados en S .
25. Combinar dos números S y T activando todos los bits de S que estén activados en T .
26. Encontrar el número X tal que S XOR X = T .
27. Encontrar la unión de dos conjuntos representados por máscaras de bits S y T .
28. Encontrar la intersección de dos conjuntos representados por máscaras de bits S y T .
29. Calcular la diferencia simétrica entre S y T .
30. Apagar todos los bits después del bit más significativo activado en S .
31. Apagar todos los bits excepto el bit más significativo activado en S .

32. Generar todos los subconjuntos de un conjunto representado por una máscara de bits S.
33. Calcular la paridad (par/impar) del número de 1s en S.
34. Encontrar el siguiente número mayor con el mismo número de 1s que S.
35. Encontrar el número anterior menor con el mismo número de 1s que S.
36. Determinar si S es un palíndromo en su representación binaria.
37. Dividir S en dos partes separando sus bits en la posición P.
38. Verificar si S contiene bits alternados (por ejemplo, 101010...).

Fenwick Trees

Para los siguientes problemas, resolver escribiendo en pseudocódigo o en C++:

1. Implementar un Fenwick Tree para manejar consultas de suma en un rango.
2. Actualizar un elemento específico en un arreglo utilizando un Fenwick Tree.
3. Calcular la suma de los primeros k elementos de un arreglo usando un Fenwick Tree.
4. Determinar el valor de un elemento individual del arreglo original utilizando un Fenwick Tree.
5. Construir un Fenwick Tree desde un arreglo dado en $O(n)$.
6. Incrementar todos los elementos en un rango $[l, r]$ en un valor x.
7. Establecer todos los elementos en un rango $[l, r]$ a un valor constante utilizando un Fenwick Tree.
8. Calcular el número de elementos menores o iguales a un valor dado x en un rango.
9. Resolver el problema de rango máximo-mínimo utilizando un Fenwick Tree.
10. ¿Cómo se implementaría un Fenwick Tree bidimensional?
11. Encontrar la posición de un valor dado en un arreglo ordenado utilizando un Fenwick Tree.
12. Determinar el k-ésimo elemento más pequeño en un arreglo utilizando un Fenwick Tree.
13. Implementar un Fenwick Tree que permita consultas de máximo en un rango.
14. Contar el número de elementos mayores o menores que un valor dado en un rango específico.
15. Responder consultas de frecuencia de un valor x en un rango dado.
16. Investigar sobre el problema del Conteo de Inversiones en un arreglo. Resolver el problema utilizando un Fenwick Tree.
17. Construir un Fenwick Tree donde cada nodo almacene el producto de los elementos en lugar de la suma.
18. Manejar consultas de suma al cuadrado de los elementos en un rango usando un Fenwick Tree.

19. Contar ocurrencias de un carácter en una cadena dentro de un rango utilizando un Fenwick Tree.
20. Implementar un algoritmo para realizar consultas de frecuencia de subcadenas en tiempo $O(\log n)$.
21. Calcular la frecuencia acumulativa de palabras en un rango de texto utilizando un Fenwick Tree.
22. Resolver el problema de "Matching Parentheses" utilizando un Fenwick Tree para manejar rangos.
23. Contar el número de elementos únicos en un rango usando un Fenwick Tree.
24. Resolver problemas de rango de diferencia absoluta mínima utilizando un Fenwick Tree.

Memoization

1. Calcular el n -ésimo término de la serie de Fibonacci usando memoization.
2. Resolver el problema de la escalera (cuántas formas hay de llegar al escalón n con pasos de 1 o 2).
3. Determinar si una cadena puede segmentarse en palabras válidas de un diccionario (Word Break Problem).
4. Calcular el coeficiente binomial $C(n, k)$ usando memoization.
5. Calcular sumas de rango utilizando memoization.