1 Problema

Sea $G(E,V)$ un árbol binario. La distancia entre dos nodos es la longitud del camino que los conecta (suponemos que la distancia entre vecinos es 1). El **diámetro** de un arbol es la distancia maxima entre dos nodos cualesquiera. Encontrar un algoritmo lineal en el número de nodos para encontrar el diámetro de un árbol.

2 Problema

Sea $X_1, X_2, \cdots, X_n$ una secuencia de números reales (no necesariamente positivos). Diseañar un algoritmo lineal que encuentre una subsecuencia de números (consecutivos) cuyo producto sea maximal. Por comodidad definimos la subsecuencia vacía con producto 1.

3 Problema

Dado un árbol binario $G(V,E)$, la distancia entre dos nodos es la longitud del camino que los conecta (suponemos que la distancia entre dos vecinos es 1). Queremos construir una matriz A cuya entrada A_{ij} sea la distancia entre los nodos v_i, v_j del rbol. Disear un algoritmo cuadrtico que construya esta matriz, cuando el rbol se entrega como la matriz de adyacencia.

4 Problema

Sea $x_1, x_2, \cdots, x_n$ un conjunto de enteros, y sea $S = \sum_{i=1}^{n} x_i$. Disear un algoritmo de complejidad $O(nS)$ que divida el conjunto de enteros en dos conjuntos de sumas iguales, o que determine que tal divisin es imposible.

5 Problema

Suponga que le dan un algoritmo X en *una caja negra*, es decir, que no sabe como esta hecho ni puede modificarlo. Dada una secuencia de nmeros reales y un nmero k, el algoritmo responde "si" o "no" dependiendo de si la secuencia tiene un subconjunto de nmeros cuya suma es exactamente k. Disee un algoritmo que utiliza $O(n)$ llamadas al algoritmo X para *encontrar* el subconjunto que suma exactamente k. (note que X solo dice si la subsecuencia existe)

6 Problema

Construya un algoritmo que encuentre el mximo y el mnimo de un conjunto. El algoritmo debe utilizar a lo ms $\frac{3n}{2}$ comparasiones (si $n = 2^k$). Establezca

la razn por la cual este algoritmo no requiere las $2n-3$ comparaciones de un algoritmo trivial de fuerz bruta.

7 Problema

Dado un arreglo de enteros $A[1 \ldots n]$, tal que, para todo i, $1 \leq i < n$. Se cumple que $|A[i] - A[i+1]| \leq 1$. Sea $A[1] = x$ y $A[n] = y$ y adems $x < y$. Disee un algoritmo de bsqueda para encontrar j tal que $A[j] = z$ para algun valor z acotado por $x \leq < \leq y$. Cual es el mximo nmero de compariciones que hace su algoritmo?.

8 Problema

La entrada es un conjunto S de nmeros reales. Disee un algoritmo $O(n)$ para encontrar un nmero que *no* est en el conjunto S. Pruebe que la cota mnima del nmero de pasos para resolver el problema es $\Omega(n)$.

9 Problema

La entrada es un conjunto S de nmeros reales, y un nmero x. Disee un algoritmo para determinar si hay k elementos ($k \geq 2$) de S cuya suma sea exactamente x. El algoritmo debe correr en $O(kn \log n)$ time.

10 Problema

Dados dos conjuntos S_1,S_2, y un nmero real x, encontrar dos elementos uno de S_1 y otro de S_2 tales que su suma es exactamente x. El algoritmo debe correr en tiempo $O(n)$, cuando n es el nmero total de elementos en ambos conjuntos.

11 Problema

Dados dos conjuntos, disear un algoritmo que permita determinar si los dos conjuntos son disjuntos. Dar la complejidad del algoritmo en trminos de los tamaos m y n de los dos conjuntos. Asegurese de considerar el caso de cuando uno de los dos conjuntos es significativamente menor que el otro.

12 Problema

La entrada es una secuencia de nmeros reales $x_1, x_2, \cdots, x_n$ con n par. Disee un algoritmo que divida el conjunto en $n/2$ pares de la siguiente manera: Para cada par calculamos la suma de sus componentes. Denotamos por $s_1, s_2, \cdots, s_{\frac{n}{2}}$

las $n/2$ sumas. El algoritmo debe encontrar la particin que minimiza el mximo de las s_i.

13 Problema

La entrada son d secuencias de elementos, cada una de ellas ordenadas. El número total de elementos es n. Disear un algoritmo $O(n \log d)$ para mezclar todas las secuencias para obtener una secuencia final ordenada.

14 Problema

La entrada es una secuencia de n enteros con muchas duplicaciones, de tal modo que el número de elementos *distintos* es $\log n$ y el número total de elementos es n. Disee un algoritmo para ordenar este conjunto utilizando *a lo mas* $O(n \log \log n)$ comparaciones en el peor caso.

15 Problema

La entrada es un heap de tamao n (en el que el elemento mas grande está en el tope) y un número real x. El heap está dado como arreglo. Disear un algoritmo para determinar si el k-ésimo elemento mas grande en el heap es menor o igual a x. El peor caso de este algoritmo debe ser $O(k)$, independientemente del tamao del heap. Se puede utilizar $O(k)$ espacio auxiliar. (Note que no necesita encontrar el $k$ésimo elemento mas grande, solo necesita determinar su relación de igualdad con x).

16 Problema

Sea A un algoritmo que encuentra el k-ésimo elemento mas grande utilizando una sucesión de comparaciones. Probar que A posee suficiente información para determinar cuales elementos son mas grandes que el k-ésimo mas grande y cuales son mas chicos que el k-ésimo mas grande. En otras palabras, es posible partir el conjunto alrededor del k-ésimo sin hacer mas comparaciones que las que ya hizo A.

17 Problema

Una secuencia S binaria ciclica de tamao 2^n es llamada de "De Bruijn" si *toda* secuencia binaria s de tamao n se encuentra como subsecuencia de S. Por ejemplo, para $n = 3$ la secuencia 10001011 es de "De Bruijn". Disee un algoritmo que genere una secuencia de "De Bruijn" para una n dada.