Gran Premio Mexicano

Análisis del Concurso 1

HaKings

Alfredo Altamirano y Diego Gutierrez Yepiz

A - Acronyms

- **Problema:** Dados tres nombres y tres palabras, checar si el acrónimo formado por las tres palabras pudo haber venido de las iniciales de los nombres.

- **Solución**: Checar si son iguales la primer letra del nombre y la primer letra de la palabra correspondiente.

B - Tobby Bones

- **Problema**: Dada una secuencia de números:

  Contestar queries (i, j, S) con el numero de de elementos de la secuencia de los índices i a j con valor menor o igual a S.
  Actualizar el elemento i con el valor S.

- **Solución**: Segment tree con un árbol binario de búsqueda balanceado en cada nodo.
  El ABB de cada nodo contiene los elementos de la secuencia cubiertos por ese nodo.
  El ABB debe poder contestar el índice de algún número en el ABB.
  Para actualizar la posición i, se modifican el ABB de todos los nodos que cubren la posición i (a lo mucho logN nodos).
  Para contestar la query, se encuentran los nodos que cubren completamente el rango [i, j] (a lo mucho logN nodos), y en cada uno se cuentan los elementos menores a S.

B - Tobby Bones

- **Hint**:

  tree<int> es como un set, pero permite saber el índice el un elemento.

  Segment Trees: http://codeforces.com/blog/entry/18051

  ABB con kth order statistics: http://codeforces.com/blog/entry/11080

C – Tobby and Query

- **Problema**: Dado N, números responder queries en el rango [l,r]
- **Solución**: Cualquier estructura que maneje queries en rangos es válida, por ejemplo Segment tree, sin embargo se puede notar que las queries son offline, entonces se pueden usar sumas parciales.
- **Hint**: Si se usan bitwise operations en el segment tree se puede reducir la complejidad de tiempo a logN por query

D - Standard Deviation

- **Problema**: Dado N, calcular la desviación estándar de los primeros N números impares.
- **Solución**: Precalcular la suma de los cuadrados de los primeros 10^6/2 números pares y precalcular la suma de los primeros 10^6/2 números impares. Encontrar el promedio en tiempo constante (es igual a la mediana en este caso). La desviación estándar es la raíz de un número x/n-1. x es dos veces la suma de los cuadrados de los primeros k números pares o impares (depende de si el promedio es par o impar).
- **Hint**: Usar long long

E - Tobby and the LED display

- **Problema**: Rotar un string de tamaño N a la derecha o a la izquierda T veces.
- **Solución**: Obtener el string sin las separaciones.
  Si inicialmente el string empezaba en el índice 0, ahora empieza en T%N si se rota a la izquierda, o en (T%N+N)%N si se rota a la derecha.
  Imprimir el string empezando en el nuevo índice y usando las separaciones.

F – Triangular Test II

- **Problema**: Dado N cual es la menor cantidad de números triangulares que se necesitan para que al sumarlos su respuesta sea N.
- **Solución**: Usar un BFS y meter a la queue solo aquellos elementos que no han sido visitados con el costo del nodo actual+1.
  Las transiciones son todos los números triangulares tal que actual+t[i] sea menor que maxN.
- **Hint**: Solo hacer eso sería muy lento, darse cuenta de cuál será la mayor respuesta y matar la queue ahí.

G – Tobby and the line game

- **Problema**: Dados punto inferior izquierdo y superior derecho de un rectángulo encontrar el valor esperado del cuadrado de la distancia al tirar 2 monedas (todos los puntos tienen la misma probabilidad de ocurrir, incluso en posiciones reales).

- **Solución**: Aparentemente muchas personas solo identificaron la formula y la hardcodearon, es válido en cuestiones de concurso pero no siempre funciona, aquí va la solución correcta.
  Darse cuenta que el hecho de que la fórmula de distancia esté al cuadrado permite tratar independientemente las xs de las ys, es decir no importa tu valor en xs tu valor esperado en y es siempre el mismo. Así que esto permite partir el problema y resolverlo en 1 dimensión (primero las xs y luego las ys) y la respuesta será la suma de ambas. La manera de obtener valor esperado cuando se trata de reales es integrando (para discretos se hace formula de sumatoria).

G – Tobby and the line game

- **Solución Cont…**
  Primero simplifiquemos, traslademos los puntos para que la esquina inferior izquierda se encuentre en el 0,0. Resolveremos solo para las xs, para las ys es exactamente igual.
  Después supongamos que estamos en una posición, llamémosla P, el costo esperado en P seria el valor esperado desde 0 hasta P y desde P hasta xr, en otras palabras y por supuesto que como todas tienen la misma probabilidad tenemos que obtener la probabilidad de que caiga en el primer rango (P/xr) y la que caiga en el segundo ((xr-p)/xr) y para cada rango dividimos entre la cantidad de valores (el promedio de cada rango)

G – Tobby and the line game

- **Solución cont...**

$$\frac{P * \int_0^P x^2}{xr * P} + \frac{(xr - P) * \int_0^{xr-P} x^2}{xr * (xr - P)} = \frac{\int_0^P x^2}{xr} + \frac{\int_0^{xr-P} x^2}{xr} = \frac{P^3}{3 * xr} + \frac{(xr - P)^3}{3 * xr}$$

Una vez que tenemos esto es importante darse cuenta que el valor P puede tener cualquier valor desde 0 hasta xr, entonces de igual manera integramos y necesitamos el promedio (volvemos a dividir entre xr)

$$\frac{\int_0^{xr} P^3}{3 * xr^2} + \frac{\int_0^{xr} (xr - P)^3}{3 * xr^2}$$

G – Tobby and the line game

- **Solución cont...**

  Expandimos y resolvemos y se simplifica a:

$$\frac{xr^2}{6}$$

H - Painting the Wall

- **Problema**: Dada una matriz indicando las posiciones que deben ser pintadas y las que no, encontrar el menor número de líneas verticales y horizontales para pintar la matriz.
- **Solución**: Crear grafo bipartito donde los nodos del lado izquierdo son las posibles líneas verticales y los nodos del lado derecho son las posible líneas horizontales a pintar. Crear arco para cada par de líneas (x, y) tal que el no pintar x implique que y debe ser pintado para cubrir la matriz. Encontrar el minimum vertex cover del grafo (es igual al maximum bipartite matching)
- **Minimum Vertex Cover in bipartite graph**: https://en.wikipedia.org/wiki/K%C5%91nig%27s_theorem_(graph_theory)
- **Maximum bipartite matching**: http://www.geeksforgeeks.org/maximum-bipartite-matching/

I - Tobby on Tree

- **Problema**: Dado un árbol con valores en cada nodo, contestar queries (a, b) con el máximo común denominador de los valores de los nodos en el camino de a a b. También actualizar el valor de un nodo.
- **Solución**: Heavy light decomposition con segment trees.
- **Heavy Light Decomposition**: http://blog.anudeep2011.com/heavy-light-decomposition/
- **Segment Trees**: http://codeforces.com/blog/entry/18051

J – Tobby Stones

- **Problema**: Se tiene una fila de N piedras, originalmente todas son blancas y hay 4 tipos de Queries, cuantas piedras negras y blancas hay en el rango [l,r], revertir el orden en un rango [l,r], invertir los colores en un rango [l,r], cambiar todas las piedras en un rango [l,r] al mismo color, ya se blanco o negro.

- **Solución**: Un segment tree con lazy propagation resuelve 3 tipos de queries, pero no la de revertir el orden en un rango, así que no funciona, se necesita algo más. Es necesario usar un cartesian tree (también conocido como Treap). Es una estructura de datos que nos permite realizar todas esas operaciones en tiempo logN en promedio por query.

J – Tobby Stones

- **Hint**: Por la manera en que funciona un Cartesian tree los 4 tipos de queries se pueden hacer muy fácilmente, solo hay que poner atención al método push para que los cambios se propaguen en el árbol de manera correcta.

K – Tobby and Seven

- **Problema**: Dado N y K posiciones de bits, cual es el mayor número que se puede formar que sea divisible entre 7. Se pueden cambiar los bits en cualquiera de las k posiciones por otro bit en cualquier otra de las k posiciones.
- **Solución**: K vale máximo 20, entonces en el peor caso hay 20 en 10 permutaciones, que es 184,756, así que fácilmente podemos iterar sobre todas ellas y para cada una verificamos si el resultado es divisible entre 7, si lo es agarramos el mayor de todos esos. Por cada iteración son K pasos, así que en el peor caso serán menos de 4*10^6, lo cual fácilmente corre en tiempo.
- **Hint**: Usar next_permutation.

L – Tobby and Prime Sum

- **Problema**: Dados L y R, cuantos números existen tal que la suma de sus dígitos sea un número primo en el intervalor [L,R]. Imprimir la respuesta módulo 10^9+7.

- **Solución**: La respuesta será la solución de R – la de L – 1. L y R valen máximo 10^500, así que podemos notar que la suma de los dígitos va a ser un número menor a 5000, entonces podemos usar un DP[N] [sum], donde N es cuantos dígitos nos quedan y sum es el acumulado de la suma hasta este punto.

L – Tobby and Prime Sum

- **Solución cont…**
  Termina siendo 10*500*5000=2.5*10^7 por ejecución del programa (no por caso, ya que podemos utilizar lo que teníamos en el caso pasado para el siguiente). Por supuesto que ese DP es suponiendo que podemos usar cada uno de los 10 dígitos en la siguiente posición, así que falta considerar los casos en los que no podemos usar todos los dígitos, por ejemplo en 46, solo podemos usar hasta el 4 en las décimas y hasta el 6 en las unidades, dado que la décima es 4. Para resolverlo iteramos sobre cada dígito de L y R y vamos obteniendo la respuesta parcial y vamos sumando (aplicando módulos).