CONFUNDIENDO PALABRAS

AUTOR: LAUTARO LASORSA

Solución Esperada

Para la solución esperada son necesarias una serie de observaciones.

Primero, como solo nos interesan los pares de palabras que tienen el mismo largo y difieren en una letra, podemos dividir las palabras según su longitud, resolver el problema independientemente para cada conjunto de palabras de la misma longitud, y finalmente sumar las respuestas para obtener la respuesta al problema completo.

Segundo, que 2 palabras difieran en una sola posición implica que si eliminamos esa posición las 2 palabras son iguales. O, lo que es lo mismo, si s1 y s2 difieren en la posición i, **s1[0:i]+s1[i+1:|s2|] = s2[0:i]+s2[i+1:|s2|]**, pero podemos descomponer esta igualdad en que el prefijo que termina en i debe ser igual **(s1[0:i]=s2[0:i])** y el sufijo que empieza en i+1 debe ser igual **(s1[i+1:|s1|]=s2[i+1:|s2|])**. Por tanto, nos interesa una forma de hacer esas dos comparaciones rápidamente.

Tercero, utilizando un **Trie** podemos asignarle a cada prefijo y cada sufijo de cada palabra un id en **O(M)**, de tal forma que 2 prefijos solo tendrán el mismo id si son iguales, y lo mismo para dos sufijos. Para los prefijos la forma es obvia, ya que insertar las palabras en el **Trie** genera para cada prefijo un id (el número de nodo que representa ese prefijo). Lo importante es notar que, si la insertamos al revés, estamos generando un id para cada sufijo.

Finalmente, lo que nos queda ver es para cada posible posición i, cuáles palabras se confunden al eliminarla, pero con los ids antes generados podemos, para cada i, definir a cada palabra como el par ordenado **{prefijo[i],sufijo[i+1]}** (el prefijo que termina en i sin incluirla, y el sufijo que empieza en i+1), y ahora podemos agruparlos según su par ordenado por ejemplo utilizando un map.

Ahora el problema se reduce a, para cada bolsa, saber la suma de los valores absolutos de las diferencias. Una posible forma es ordenando los enteros y recorrerlos de menor a mayor. En cada paso, como el elemento que estamos viendo **(v)** es mayor o igual a todos los anteriores **(i elementos que suman s)**, podemos saber la

diferencia haciendo únicamente **(v*i)-s** (ver que es (v-elem[0])+(v-elem[1])+...+(v-elem[i-1])).
Los pasos del map y el ordenamiento tiene complejidad **O(N)**, y en total se realizan **M** inserciones en maps y se ordenan vectores que tienen en total **M** elementos. Por tanto, la complejidad total es **O(M*log(N))**

Notar que si se ordenan todas las palabras según su valor desde un principio usando counting sort (**O(maxV)**) y luego se utiliza un **unordered_map (tabla hash)** para hacer las bolsas de palabra se puede lograr una complejidad de **O(M+maxV)**.

Por otro lado, es importante tener presente que si se observa que para un largo L fijo, N*L=M, y por tanto o N<= $M^{1/2}$ o L<=$M^{1/2}$ permite realizar soluciones O($M^{3/2}$) aplicando soluciones O(N^2*L) u O(N*L^2) según sea el caso. El si estás soluciones son aceptadas o no va a depender de la robustez de los casos de prueba, el tiempo límite y las cotas finales que tenga el problema.

Soluciones Parciales

Para la primer subtarea **(N, max(palabras[i]) <=100)** se espera que para cada posible par de palabras (hay **O(N^2)**) y para cada posición (si asumimos que tienen el mismo largo, y este es **L**, hay **L** posiciones) vean en tiempo lineal (**O(L)**) si difieren en esa posición o no. Por tanto, la complejidad puede ser hasta **O(N^2*L^2)** y entrar en tiempo.

Para la segunda subtarea **(N, max(palabras[i]) <=500)** la idea esperada es que compare como antes cada par de palabras **(O(N^2))** pero que en una sola comparación con costo **(O(L))** detecte la cantidad de posiciones en las que difieren. Eso da una complejidad **O(N^2*L)**.

Por otro lado, para esta misma subtarea es posible una solución **O(N*L^2 + L*N*log(N))** si para cada posible posición a saltearse se pone a todas las palabras en un trie y luego se resuelven las bolsas de palabras iguales como en la solución esperada total.

La tercera subtarea, donde todas las palabras tienen la misma longitud, espera que realice lo que en la solución global hace para aplicar por separado a cada conjunto de palabras de igual longitud.

CONFUNDIENDO PALABRAS

AUTOR: LAUTARO LASORSA

En la cuarta subtarea se espera que realice algo igual a la solución global salvo por cómo resolver la suma de las diferencias entre palabras que solo varían en una letra. En este caso como solo hay 2 tipos de valores y una confusión entre ellos cuesta uno el costo es el producto de las cantidades que hay de uno y otro tipo.