CC4102 **Diseño y Análisis de Algoritmos** **2021A**

Jérémy Barbay, Javier Oliva

Ejercicio Formativo 2- Reentrega

(Fecha de entrega: [2021-04-25 Tue])

Sofía Valentina Bobadilla Ponce

1 Definiciones previas

En este caso es fundamental realizar observaciones:

- A pesar de ser el peor caso , el *guesser* terminará la partida descifrando la palabra escondida por *setter*.
- El *guesser* se verá forzado a utilizar el máximo numero de preguntas (*guesses*), por ello en la siguiente solución se dirá cuál sería el orden del máximo de preguntas que tendría que hacer para descifrar el problema.
-

1. Suppose that D = $[0..\sigma - 1]^k$ for some integers σ and k: what is the worst case complexity of the *guesser* for σ and k fixed? Justify your answer! Hint: Design an adversary strategy for the setter in order to force the guesser to performs the maximum number of guesses.

   Con lo escrito en las definiciones previas se determina que el peor caso será cuando el *Evil Hangman* escoga un diccionario D tal que existan muchas cadenas de caracteres similares con el fin de cambiar su palabra la mayor cantidad de veces, en particular podría negar todos los caracteres hasta llegar a una cadena de largo k compuesta por un único tipo de carácter.Cada vez que el *guesser* escoge un carácter el *setter* divide el conjunto D entre aquellas que contienen el caracter y aquellas que no lo son. Para poder ocupar el máximo de *guesses* se utilizaran las g tal que con cada una de ellas para la adivinanza g-1 se reduzca el conjunto D a dos palabras tal que con la adivinanza g se determina la palabra.
   Viendo lo anterior como un árbol desequilibrado donde cada adivinanza i determinará las hojas.En total se tienen σ^k palabras o cadenas en el conjunto D . El peor caso será tal que con la palabra finalmente escogida estará compuesta por el último caracter a preguntar , es decir se harán σ consultas , con ello el orden sería O(n) con n =σ = cantidad de caracteres.Un símil a esto sería cuando al buscar en un arreglo en contra un adversario no preguntamos por todos los espacios y el adversario podría escoger justamente ese espacio por el que no se consulta.

2. . Suponga que D $\subseteq [0...\sigma 1]^k$ es de tamaño n, para σ, k, n enteros: ¿cuál es la complejidad de peor caso para el *guesser* con σ, k y n fijos? Justifique su respuesta. A diferencia del caso anterior en esta ocasión el conjunto de no se compone de las cadenas formadas de la combinación de los σ caracteres, en particular se tiene :

   - tamaño del conjunto D =n
   - n caracteres = σ
   - tamaño cadenas = k

   A pesar de las diferencias entre esta pregunta y la primera , la estrategia del adversarió será la misma sin importar si σ, k y n fijos ya que el conjunto D que se menciona es un subconjunto del diccionario D en la parte 1 , y el tamaño de este seguirá siendo σ , con esto para que el *guesser* encuentre la palabra debe pagar el costo que esto conlleva , ya que al igual que en la pregunta anterior el *setter* lo forzará a pagar el mayor costo posible, es decir, el *guesser* debe adivinar todas las letras lo cual nuevamente tiene complejidad $O(\sigma)$.

3. ¿Qué diferencias hay entre
   • la complejidad de peor caso para el *guesser* si σ, k y n están fijos,
   • la complejidad de peor caso para el *guesser* si D está fijo y
   • la complejidad de peor caso para el *guesser* para una palabra específica w $\in$ D?

   A pesar de la diferencia entre los tres casos , y por lo visto en la parte a y b se define que utilizando la estrategia del adversario para el peor caso el *guesser* siempre se verá obligado a recorrer todas las letras del diccionario , por ello en los tres escenarios el peor caso tiene complejidad $O(\sigma)$ para su respectivo $\sigma = |D|$.