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

Jérémy Barbay, Javier Oliva

Ejercicio Formativo 2

(Fecha de entrega: [2021-03-30 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 , 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

   Para este caso no se sabe que palabras estarán específicamente en D y con ello no se sabe si estarán presenten todos los caracteres por esto el orden cambiará respecto a la pregunta anterior.Para esto se analiza un arbol que inicialmente tiene al conjunto D de tamaño n , con cada $guess_i$ el conjunto se verá dividido entre dos familias una que cumpla la pregunta y otra familia que no , así hasta que el adversario no pueda seguir cambiando la palabra y el *guesser* adivine la palabra.Esto tendría un costo de O(log(n)) ya que el costo de la división es $O(\log_2(n))$.

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?

   Entre los primeros dos la diferencia está en que con σ , k y n fijos se puede obtener mucha mas información y por ende el peor caso sería *mejor* que para cuando unicamente D está fijo. Similar con la tercera consulta , el tener una palabra específica acotara los escenarios a evaluar.