

Proyecto Análisis de Algoritmos - Picas y Fijas

Adrián García y Nicolás Miranda

November 2019

1. Descripción del problema

Dado un alfabeto, se genera aleatoriamente una secuencia de u caracteres del alfabeto, secuencia que puede denominarse R. Luego, se genera también aleatoriamente otra secuencia con u caracteres del alfabeto, que puede denominarse S.

Después, R y S se comparan usando una función conocida como juez. Un elemento de R y S que sea igual y ocupe la misma posición en ambas secuencias, se conoce como fija. Un elemento de R y S que sea igual pero no ocupe la misma posición en ambas secuencias se conoce como pica. Por tanto, si el número de fijas es igual a u , S y R son iguales. El retorno de la comparación de S y R (la función juez) retorna el número de fijas y el número de picas.

El problema radica en ir modificando la secuencia S hasta que ésta sea igual a R, haciéndolo en la menor cantidad de pasos/tiempo posible (por iteraciones), conociendo: el alfabeto dado, el número u de caracteres, los retornos de número de picas y número de fijas brindados por la función juez en cada iteración, así como conociendo cómo se ha ido modificado la secuencia S también en cada iteración.

Dicha modificación se realiza mientras el número de fijas sea menor a u . Si el número de fijas en la primera iteración es igual a u , la modificación de S no es necesaria (la primera iteración es dada por la comparación de R y la secuencia S inicial).

2. Formalización

Enseguida se presenta el diseño del algoritmo responsable de la modificación de la secuencia S. Cabe resaltar que las funciones encargadas de generar y comparar las secuencias R y S no son descritas aquí, sino que se usan de manera complementaria fuera del algoritmo cuyo diseño se presenta en este documento.

2.1. Entradas

- Un alfabeto \mathbb{T} .
- Un número $n \in \mathbb{N}$, donde n es el tamaño del juego a adivinar, es decir, el número de posiciones que tendrá el juego.
- Dos números $b, c \in \mathbb{N}$, donde b es el número de fijas y c es el número de picas encontradas en un turno dado, tal que $b + c \leq n$ para cualquier iteración i dada.

2.2. Salidas

- Una secuencia A de u caracteres, con $A = \langle a_i \in \mathbb{T} | a_1, \dots, a_u \rangle$ y $a_1 \neq \dots \neq a_u$, donde \mathbb{T} es el alfabeto provisto como entrada.

3. Tipo y clase de problema

El **tipo de problema** es de decisión porque se toma la decisión de cambiar o mantener un carácter del alfabeto en la solución. La **clase de problema** es NP-completo, porque el algoritmo de fuerza bruta es

superpolinomial, en este caso de picas y fijas, se requiere hacer permutaciones sin repeticiones para hallar todas las combinaciones posibles que satisfagan la solución, además, el algoritmo de verificación es polinomial.

4. Heurística

Antes de mencionar los pasos que tiene la heurística, es oportuno hacer hincapié en los casos o condiciones que se pueden presentar y que determinan las decisiones que se deben tomar en el siguiente juego. Los casos son los siguientes:

- Cuando tanto el número de picas como el número de fijas es igual a cero.
- Cuando el número de fijas más el número de picas es igual al tamaño del juego a adivinar.
- Cuando no se cumplen ninguna de las dos condiciones descritas anteriormente.

La heurística para la solución del problema de picas y fijas se presenta a continuación:

1. Si se presenta el caso en el que el número de picas y el número de fijas sea igual a cero, entonces significa que ninguno de los elementos que están en el juego no se encuentran dentro de la solución y por lo tanto se procede a eliminar los elementos del alfabeto con el fin de reducir el tamaño del mismo.
2. Si la condición del paso 1 no se cumple, entonces se procede a revisar si el número de fijas más el número de picas es igual al tamaño del juego a adivinar, entonces se procede a igualar el alfabeto inicial con el alfabeto del juego actual con el fin de sólo realizar modificaciones con el alfabeto encontrado, ya que todos los elementos que están en el juego son los que están dentro de la solución y por lo tanto, se empezará a mover los elementos que son picas con el fin de convertir estos en fijas. Una vez se entra a este caso, se revisa cuáles elementos son picas y se para cada una se empieza a intercambiar la posición con otras picas hasta convertirla en fija.
3. Si la condición del paso 1 no se cumple y la condición del paso 2 tampoco se cumple, entonces se procede a ir cambiando de cada uno de los elementos de la solución en ese momento, con el fin de encontrar una pica o una fija y de esta forma, poder llegar posteriormente al paso 2 de la heurística para simplificar la solución. Cuando se realiza el cambio se revisa si el número de picas y de fijas cambió respecto a número de picas y fijas anterior y si el cambió se dan las siguientes posibilidades:
 - a) Si el número de fijas del intento actual es mayor al número de fijas del intento anterior, significa que lo que se acabo de agregar es una fija y se marca la posición para conocer que ahí existe una fija.
 - b) Si el número de fijas del intento actual es menor al número de fijas del intento anterior, significa que el elemento en la posición que se modificó es una fija y por lo tanto se revierte el cambio y se marca la posición de la fija.
 - c) Si el número de picas del intento actual es mayor al número de picas del intento anterior entonces se guarda la posición de la nueva pica.
 - d) Si el número de picas del intento actual es menor al número de picas del intento anterior entonces se revierte el cambio y se marca la posición de la pica.

El objetivo de la heurística es realizar cambios para detectar fijas y picas y una vez detectadas todas ellas, se convierten las picas en fijas.

5. Complejidad

El algoritmo propuesto tiene una complejidad de $O(n)$ ya que se recorre de manera lineal, tanto la solución como el arreglo que informa donde se encuentran las fijas.

6. Invariante

La invariante del algoritmos consiste en que en la variable *posicionesFijas* es guardado con un valor booleano las fijas detectadas mediante los cambios realizados por la heurística desarrollada.

7. Cambios realizados respecto a la primera entrega

Los cambios que se realizaron en el diseño de la segunda entrega con respecto al diseño de la primera entrega fueron los siguientes:

- En la segunda entrega ya no se tiene un número $n \in \mathbb{N}$, que indica la cantidad de iteraciones que han sido realizadas hasta cierto momento, ya que se determinó que no es relevante para la solución de este tipo de problema de decisión.
- El número $u \in \mathbb{N}$ de la primera entrega, es equivalente al número $n \in \mathbb{N}$ de la segunda entrega.
- Para la primera entrega se había pensado utilizar una matriz *historialSecuencias* de u filas y n columnas, la cual iba guardando todas las secuencias que se iban generando en cada una de las iteraciones, sin embargo, para la segunda entrega se descubrió que no era eficiente ya que consumía bastantes recursos y no era útil para los fines del algoritmo planteado.
- Para la primera entrega se había pensado utilizar dos arreglos unidimensionales *historialFijas* e *historialPicas*, donde se iba guardando el número de picas en cada una de las iteraciones, sin embargo, para la segunda entrega se decidió no utilizar estos parámetros porque no eran necesarios de acuerdo a los requisitos del proyecto y la idea es tomar la decisión basándose en los resultados de picas y fijas inmediatamente anteriores.