Parcial 2020 2C 1R

1) Recorremos todos los documentos y armamos un archivo auxiliar con los términos y el documento donde se encuentra dicho termino. En nuestro caso el archivo sería así

- Caballo->1
- Caballero->1
- Cabra->1
- Cacao->2
- Cacerola->2
- Caballo->2
- Cabalgar->2
- Cacao->3
- Cabra->3
- Cabaña->3
- Caballo->4
- Cabalgar->4
- Caballero->4
- Caballo->4
- Cabra->5
- Cacerola->5

Luego ordenamos dichos terminos por término y documento. En nuestro caso quedaría de la siguiente forma

- Cabalgar->2
- Cabalgar->4
- Caballero->1
- Caballero->4
- Caballo->1
- Caballo->2
- Caballo->4
- Cabaña->3
- Cabra->1
- Cabra->3
- Cabra->5
- Cacao->2
- Cacao->3
- Cacerola->2
- Cacerola->5

Unificamos los términos

- Cabalgar->2,4
- Caballero->1,4
- Caballo->1,2,4
- Cabaña->3
- Cabra->1,3,5
- Cacao->2,3
- Cacerola->2,5

Codificamos los documentos a distancias

- Cabalgar->2,2
- Caballero->1,3
- Caballo->1,1,2
- Cabaña->3
- Cabra->1,2,2
- Cacao->2,1
- Cacerola->2,3

Con esto podemos armar el índice. Armamos el léxico con front coding parcial (n=3).\ Recordemos que front coding parcial evita reconstruir mas de n palabras en una lectura de índice\ El puntero de cada término apunta a los términos concatenados en disco.

iguales	distintos	punteros terminos
0	8	0
5	4	8
6	1	12
0	6	13
3	2	19
2	3	21
0	8	24

Cabalgar	lero	o	Cabaña	ra	cao	Cacerola
0	8	12	13	19	21	24

Resta armar el índice de documentos. Para ello vamos a codificar los documentos de cada término en Gamma. Recordar que ya estaban como distancias.\ Ejemplo de codificación en gamma. g(1) = 1, g(2)=010 , g(4) = 00100, etc

010010	1011	11010	011	1010010	0101	010011
0	6	10	15	18	25	29

Unificando las estructuras quedaría

iguales	distintos	punteros terminos	punteros documentos
0	8	0	0
5	4	8	6
6	1	12	10
0	6	13	15
3	2	19	18
2	3	21	25
0	8	24	29

Cabalgar	lero	o	Cabaña	ra	cao	Cacerola
0	8	12	13	19	21	24

010010	1011	11010	011	1010010	0101	010011
0	6	10	15	18	25	29

Para calcular el espacio utilizado hay que definir los tamaños para los punteros, los valores de iguales y distintos. En este caso usamos 8 bytes (x64) para todos esos valores\ Tenemos por un lado la tabla de 7x4. Entonces usamos 7x4x8 bytes = 224 bytes.\ Para los terminos utilizamos 1 byte por caracter, entonces usamos 32 bytes.\ Y para los documentos codificados utilizamos bits, en total usamos 35 bits, redondando a bytes usamos 64 bits = 6 bytes. En total usamos 224 + 32 + 6 = 262 bytes.

2) Vamos a utilizar el índice invertido para resolver la consulta Q="Caballo Caballero" con búsqueda binaria

posicion	iguales	distintos	punteros terminos	punteros documentos

posicion	iguales	distintos	punteros terminos	punteros documentos
0	0	8	0	0
1	5	4	8	6
2	6	1	12	10
3	0	6	13	15
4	3	2	19	18
5	2	3	21	25
6	0	8	24	29

Cabalgar	lero	o	Cabaña	ra	cao	Cacerola
0	8	12	13	19	21	24

010010	1011	11010	011	1010010	0101	010011

Caballo

- Pos 3 (Cabaña) (1 indice, 1 disco) Leemos 6 char: Cabaña. Como Caballo<Cabaña, busco en la mitad de arriba.
- Pos 1 (Caballero) (1 indice). 5 iguales 4 distintos. Caballo>Caballero. Mitad de abajo
  - Pos 0 (1 indice, 1 disco) Leo 5 char: Cabal
  - Pos 1 (1 disco) Leo 4 char: lero
- Pos 2 (Caballo) (1 indice). 6 iguales 1 distinto
  - Pos 1. Ya se que ahí está caballero. Me quedo con los char: caball
  - Pos 2 (1 disco) Leo en la dir 12 el char o

Caballo es lo que buscaba, leo en indice en la posición siguiente para saber el largo del puntero de caballo (1 indice) y tiene largo 5.\ Leo el puntero a documentos en la posición 10 y obtengo 11010 en gamma.(1 disco) \ Que es 1,1,2 codificado en distancias que es 1,2,4.

Caballero

- Pos 3 (Cabaña) (1 indice) Ya lo habiamos leído. Como Caballero<Cabaña, busco en la mitad de arriba.
- Pos 1 (Caballero) (1 indice). Ya lo habíamos leído también

Caballero es lo que buscaba, leo en indice en la posición siguiente para saber el largo del puntero de caballero (1 indice) y tiene largo 4.\ Leo el puntero a documentos en la posición 6 y obtengo 1011 en gamma.(1 disco) \ Que es 1,3 codificado en distancias que es 1,4.

Queda ver en que documentos coinciden ambos terminos. En este caso los documentos 1 y 4.

3) Utilizaremos una nueva estructura para la referencia a documentos. En vez de guardar los documentos solamente, guardaremos el documento, frecuencia y posiciones.\

Caballo que se encuentra en los documentos 1,2 y 4 (1,1,2 en distancias). En el documento 4 se encuentra 2 veces.\ Por lo tanto la codificacion de caballo sería 111 113 2213. Esta última está codificado en distancias para las posiciones. Siendo que las posiciones empiezan en 1.\ Se almacenaria estos valores en gamma, lo cual el puntero de caballo apuntaría a esta codificacion 111 11011 0100101011.\ Similarmente para caballero, 112 313 -> 11010 0111011

Finalmente, se hace una búsqueda como antes, pero ahora además de hacer un and comparamos las posiciones de los terminos en el documento.\ En el documento 1, caballo está en la posición 1 y caballero en la posición 2. Es un resultado ya que la posicion de caballero está inmediatamente despues de caballo\ En el documento 4, caballo está en la posición 1 y 4 , caballero en la posición 3. No es un resultado porque caballo no está en la posición 2