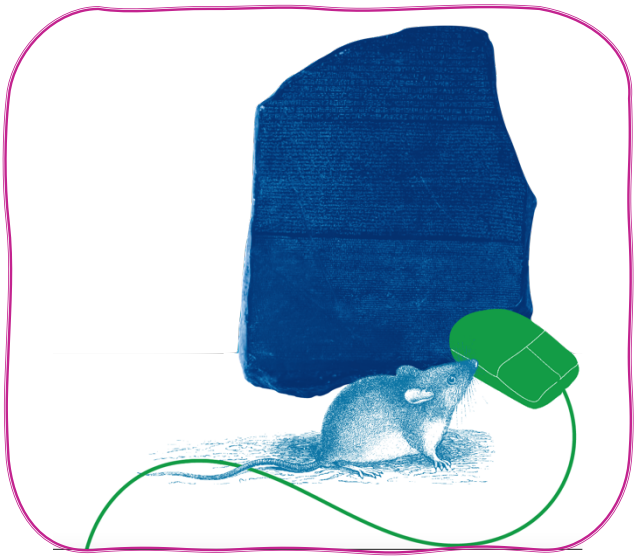
UNIVERSIDADE DA CORUÑA

FACULTAD DE INFORMÁTICA

MUBICS

Cuaderno de prácticas

Estructuras de datos y algoritmia para secuencias biológicas



Pedro Sánchez García

Curso 2021-2022

Profesor:
Dr. Fernando Silva Coira

PRÁCTICA 2. INDEXACIÓN DE SECUENCIAS

OBJETIVO

Conocer estructuras de datos y algoritmos para la compresión e indexación de secuencias.

TAREAS

A partir de un fragmento de 20 bases de una secuencia de ADN real (obtenida por ejemplo de https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi):

1. (0,15 puntos) Construir el array de sufijos correspondiente a dicho fragmento.
2. (0,15 puntos) Seleccionar un patrón cualquiera de 3 caracteres e indicar cómo se obtendrían las posiciones de sus ocurrencias en el fragmento.
3. (0,15 puntos) Construir las estructuras del array de sufijos comprimido de Sadakane, indicando los valores de las estructuras de Ψ, D y S.
4. (0,15 puntos) Usando sólo Ψ, D y S, indica cómo se obtendrían los primeros 4 caracteres del fragmento de ADN utilizado.
5. (0,15 puntos) Usando sólo Ψ, D y S, indicar cómo se contarían cuántas ocurrencias existen del patrón elegido en la tarea 1.
6. (0,1 puntos) Obtener la transformada de Burrows-Wheeler.
7. (0,15 puntos) Para el patrón elegido en la tarea 1, indicar cómo se obtendría el número de ocurrencias en el texto utilizando *backward search*.

Decido emplear la siguiente secuencia del cromosoma 1 perteneciente a la planta herbácea *Arabidopsis thaliana*:

- CCCTAAACCCTAAACCCTAA

Ejercicio 1. (0,15 puntos) Construir el array de sufijos correspondiente a dicho fragmento.

El fragmento elegido a ser indexado será T= CCCTAAACCCTAAACCCTAA$. Para ello, en primer lugar, se genera un array con sufijos del fragmento sin orden como se muestra en la tabla izquierda. Posteriormente, se procede a la construcción del array de sufijos siguiendo el orden lexicográfico de la tabla derecha, donde los sufijos del fragmento se representan mediante sus posiciones iniciales en el texto. Cabe destacar que, de esta forma, se mejora el tiempo en la realización de consultas.

SuffixArray	
0	CCCTAAACCCTAAACCCTAA$
1	CCTAAACCCTAAACCCTAA$
2	CTAAACCCTAAACCCTAA$
3	TAAACCCTAAACCCTAA$
4	AAACCCTAAACCCTAA$
5	AACCCTAAACCCTAA$
6	ACCCTAAACCCTAA$
7	CCCTAAACCCTAA$
8	CCTAAACCCTAA$
9	CTAAACCCTAA$
10	TAAACCCTAA$
11	AAACCCTAA$
12	AACCCTAA$
13	ACCCTAA$
14	CCCTAA$
15	CCTAA$
16	CTAA$
17	TAA$
18	AA$
19	A$
20	$

SuffixArray (con orden lexicográfico)	
20	$
19	A$
18	AA$
11	AAACCCTAA$
4	AAACCCTAAACCCTAA$
12	AACCCTAA$
5	AACCCTAAACCCTAA$
13	ACCCTAA$
6	ACCCTAAACCCTAA$
14	CCCTAA$
7	CCCTAAACCCTAA$
0	CCCTAAACCCTAAACCCTAA$
15	CCTAA$
8	CCTAAACCCTAA$
1	CCTAAACCCTAAACCCTAA$
16	CTAA$
9	CTAAACCCTAA$
2	CTAAACCCTAAACCCTAA$
17	TAA$
10	TAAACCCTAA$
3	TAAACCCTAAACCCTAA$

Ejercicio 2. (0,15 puntos) Seleccionar un patrón cualquiera de 3 caracteres e indicar cómo se obtendrían las posiciones de sus ocurrencias en el fragmento.

Se elige el patrón **P=AAC** para obtener las correspondientes ocurrencias en el fragmento. Para ello, se procede a realizar la búsqueda binaria sobre el array de sufijos construido en el ejercicio anterior. En la fase 1, nos dirigimos a la posición central del array de sufijos. Apreciamos que no se produce coincidencia y que el sufijo es más grande que el patrón elegido, de forma que nos desplazamos a una posición de la mitad en la región izquierda a esta, tal y como se recoge en la fase 2. En este caso, sí se produce una ocurrencia, de tal forma que analizamos la posición izquierda y derecha a esta en la fase 3. Finalmente, se llega a la fase 4, donde se analiza la derecha de la posición de la fase 3 donde existe ocurrencia. Dado que no hay ocurrencia en esta posición, se finaliza el proceso de búsqueda. De esta forma, se producen 2 ocurrencias del patrón seleccionado:

SuffixArray (con orden lexicográfico)		Fases de la búsqueda de ocurrencias
20	$	
19	A$	
18	AA$	
11	AAACCCTAA$	
4	AAACCCTAAACCCTAA$	3 (no hay ocurrencia)
12	AACCCTAA$	2 (hay ocurrencia)
5	AACCCTAAACCCTAA$	3 (hay ocurrencia)
13	ACCCTAA$	4 (no hay ocurrencia)
6	ACCCTAAACCCTAA$	
14	CCCTAA$	
7	CCCTAAACCCTAA$	1
0	CCCTAAACCCTAAACCCTAA$	
15	CCTAA$	
8	CCTAAACCCTAA$	
1	CCTAAACCCTAAACCCTAA$	
16	CTAA$	
9	CTAAACCCTAA$	
2	CTAAACCCTAAACCCTAA$	
17	TAA$	
10	TAAACCCTAA$	
3	TAAACCCTAAACCCTAA$	

Ejercicio 3. (0,15 puntos) Construir las estructuras del array de sufijos comprimido de Sadakane, indicando los valores de las estructuras de Ψ, D y S.

La construcción del array de sufijos comprimido de Sadakane implica plantear F en primer lugar, que almacena el primer carácter de cada sufijo. Posteriormente, se da forma a Ψ(i), que almacena la posición de sucesor (A[i] + 1) del sufijo representado en A[i]. Para plantear D, debemos tener en cuenta que consiste en un bitmap, donde se representa con 1 la primera aparición de un símbolo y con 0 se representa el resto. Por su parte, S es el alfabeto ordenado generado en base a F.

SuffixArray		F	Ψ	D	S
20	$	$	11	1	**$**
19	A$	A	0	1	**A**
18	AA$	A	1	0	**C**
11	AAACCCTAA$	A	5	0	**T**
4	AAACCCTAAACCCTAA$	A	6	0	
12	AACCCTAA$	A	7	0	
5	AACCCTAAACCCTAA$	A	8	0	
13	ACCCTAA$	A	9	0	
6	ACCCTAAACCCTAA$	A	10	0	
14	CCCTAA$	C	12	1	
7	CCCTAAACCCTAA$	C	13	0	
0	CCCTAAACCCTAAACCCTAA$	C	14	0	
15	CCTAA$	C	15	0	
8	CCTAAACCCTAA$	C	16	0	
1	CCTAAACCCTAAACCCTAA$	C	17	0	
16	CTAA$	C	18	0	
9	CTAAACCCTAA$	C	19	0	
2	CTAAACCCTAAACCCTAA$	C	20	0	
17	TAA$	T	2	1	
10	TAAACCCTAA$	T	3	0	
3	TAAACCCTAAACCCTAA$	T	4	0	

Ejercicio 4. (0,15 puntos) Usando sólo Ψ, D y S, indica cómo se obtendrían los primeros 4 caracteres del fragmento de ADN utilizado.

Para la obtención de los primeros 4 caracteres del fragmento de ADN, debemos partir de la primera posición de S, lo que nos conduce según ψ, a la posición 11 en D. Llevamos a cabo el recuento de 1 en D hasta la posición 11(Rank[D,11]=3), de forma que tenemos que observar la letra a la que nos conduce en S (S[2]=C). Posteriormente, nos dirigimos a la posición 14 que nos indica ψ, llevando a cabo el recuento en D, que nos conduce a C. Luego, nos situamos en la posición 17 que nos indica ψ, llevando a cabo el recuento, que también conduce a C. Se finaliza en la posición 20 marcada por ψ, donde el recuento da T, que conforma el último carácter del fragmento analizado. A continuación, se muestran las fases correspondientes junto con los resultados obtenidos:

Ψ	D	S	Etapa
11	1	**$**	**1**
0	1	**A**	
1	0	**C**	
5	0	**T**	
6	0		
7	0		
8	0		
9	0		
10	0		
12	1		
13	0		
14	0		**2**
15	0		
16	0		**3**
17	0		
18	0		
19	0		
20	0		**4**
2	1		
3	0		
4	0		

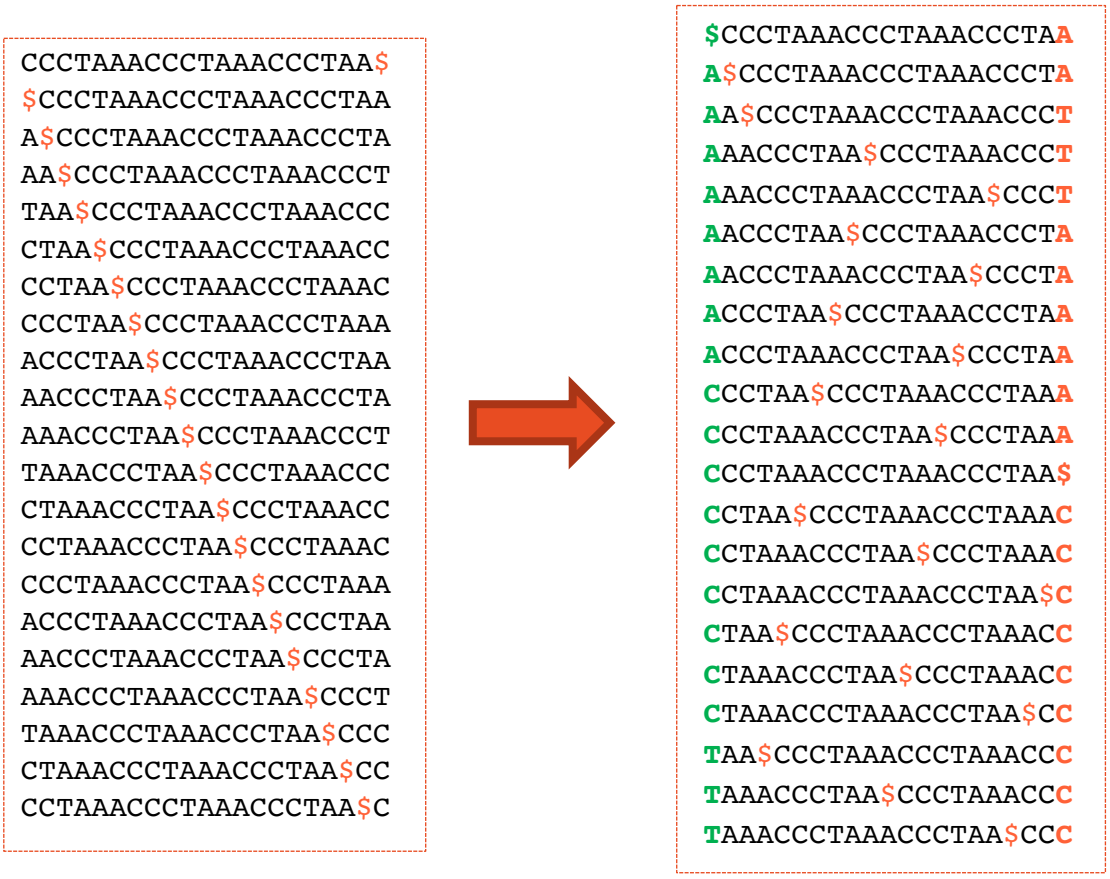
Ejercicio 5. (0,15 puntos) Usando sólo Ψ, D y S, indica cómo se contarían cuántas ocurrencias existen del patrón elegido en la tarea 1.

Para llevar a cabo el recuento de las ocurrencias del patrón **P=AAC** elegido anteriormente, se plantea la búsqueda binaria sobre el CSA. En general, se comienza en la posición 10 (etapa 1), que conduce, tras determinar rank(D,9) a descartar la región situada a la derecha de esta posición. Posteriormente, en la etapa 2, nos situamos en la posición 4, de forma que el Rank[D,4] es igual a 2 (s[1]=A), apreciamos que Ψ conduce a 6. Se acude a la posición 6 en array, calculando Rank[D,6], que me lleva a una coincidencia. Se sigue este procedimiento hasta que se alcanza una construcción de AAC tras coincidencias. Al llegar a la C, quedan como etapas en este caso la 4, donde el procedimiento nos lleva a AC, descartando la región derecha a la posición 7 en array. Por tanto, se aprecia que hay 2 coincidencias con el patrón que buscamos que son 12 y 5 en array de sufijos.

Ψ	D	S	Etapa
11	1	**$**	
0	1	**A**	
1	0	**C**	
5	0	**T**	
6	0		2
7	0		
8	0		3,4
9	0		
10	0		
12	1		1,6
13	0		5
14	0		
15	0		
16	0		
17	0		
18	0		
19	0		
20	0		
2	1		
3	0		
4	0		

Ejercicio 6. (0,1 puntos) Obtener la transformada de Burrows-Wheeler.

Para obtener la transformada de Burrows-Wheeler, se comienza con el fragmento de partida T, rotándolo progresivamente hasta generar una matriz donde las filas serán todas las rotaciones de T. Finalmente, la matriz se ordena lexicográficamente, tal y como se recoge en la siguiente figura derecha:



La última columna corresponde con la transformada de Burrows-Wheeler, que en este caso sería: AATTTAAAAAA$CCCCCCCCC.

Ejercicio 7. (0,15 puntos) Para el patrón elegido en la tarea 1, indicar cómo se obtendría el número de ocurrencias en el texto utilizando *backward search*.

Para llevar a cabo el recuento, además de la transformada de Burrows-Wheeler, debemos tener en cuenta F, que también viene dado por la región resaltada en verde para la transformada de Burrows-Wheeler. De esta forma, F será con lo que se ha trabajado en los anteriores ejercicios: $AAAAAAAACCCCCCCCCTTT. En primer lugar, se busca la última posición del patrón P=AAC, es decir, C en F. Nos fijamos en aquellas que tengan una correspondencia con A en L (flecha naranja), pues la posición anterior es A en el patrón. Observamos que hay correspondencia con A7 y A8, de forma que nos dirigimos a las posiciones de estas últimas en F (flecha verde).

Se produce correspondencia con A5 y A6 en L para ambos casos, lo que nos indica que se producen 2 ocurrencias para el patrón analizado en el fragmento, tal y como se ha determinado anteriormente en el ejercicio 2: