

Laboratorio 2

Todas las coordinaciones

FACULTAD DE INGENIERÍA

Departamento de Ingeniería Informática

Análisis de algoritmos y estructuras de datos



13/05/2020

Sopa de Letras

Este laboratorio tiene como objetivo la elaboración de un programa que permita encontrar distintas subcadenas de distintas palabras en una sopa de letras. Los datos de la sopa de letras serán entregados mediante un archivo de texto plano, cuyo nombre sera “sopa.in”, que contiene la sopa de letras de tamaño $N \times N$. Las palabras a buscar se entregarán en otro archivo de texto llamado “palabras.in”.

Las palabras pueden o no estar escritas completas en la sopa de letras. Además, pueden estar escritas de izquierda a derecha o de arriba hacia abajo. Para abordar este problema, se deben buscar **todas las subcadenas sin repetir** y almacenar en una lista simple o doblemente enlazada todas las que tengan un tamaño mayor o igual a 3. Justifique su elección de listas. Luego, se deben ordenar los substring de menor a mayor similitud. Esta similitud se define como el largo del substring correspondiente. El orden de las subcadenas del mismo tamaño debe ser en orden alfabético.

Por ejemplo, para los strings “ABABC” y “BABCA”, las subcadenas a almacenar corresponden a “ABC”, “BAB”, y “BABC”

ABABC BABCA	ABABC BABCA	ABABC BABCA
-------------------------	-------------------------	---------------------------

El formato del archivo de la sopa de letras tiene el formato especificado en la figura 1, donde la primera línea corresponde al tamaño N de la sopa, y en las siguientes líneas se encuentra el contenido de la sopa de letras. Cada línea de la sopa de letras contiene las letras de esta, separadas por un espacio.

El formato del archivo de las palabras a buscar tendrá el formato especificado en la figura 2, donde la primera línea indica la cantidad de palabras y luego en las siguientes vendrán las palabras separadas por un salto de línea.

Laboratorio 2

Todas las coordinaciones

FACULTAD DE INGENIERÍA

Departamento de Ingeniería Informática

Análisis de algoritmos y estructuras de datos



13/05/2020

20

```
u r u h y r i g a x u f q b c d p f m d
u s v y d h i a o z g e u a p b i g d c
x b i q u z q v b c k x k x t q q y r q
j l d e l c p j j e m p e o h n v k p e
b j a k e b k j u q z g u q d p e t l s
i t f t j e z n z n n l g k g v k a o y
r x s l e w l m y y n f o r j p a a w l
k q v g v t z z y o z s q g m r m r n l
d m a t i w g l n l e a s x q o v i b d
j l f p k k w p e p b v t i r o n r j l
m x v x v z i w c z m g o j v z p z k t
k e a k p b c l b o s m r j e f x d g n
a g h j g c w z j o w r r x i j y v p n
y e p f h v c k q l r u g x k d p p b t
b p x w i m k o j m a n a u f p l n d z
x c f n l l r t a l g o r i t m o s m e
s v x m c c u b h l r n b s o z g m n a
f q n q r j g e b a s y h l x n b z i j
s j t r r t w m z n r w v k b b l i f h
b a v d c q t t a r e i x q h k b l i e
```

Figura 1: Ejemplo archivo de entrada "sopa.in"

3

```
ejemplo
informatica
algoritmo
```

Figura 2: Ejemplo archivo de entrada "palabras.in"

Laboratorio 2

Todas las coordinaciones

FACULTAD DE INGENIERÍA

Departamento de Ingeniería Informática

Análisis de algoritmos y estructuras de datos



13/05/2020

20

u	r	u	h	y	r	i	g	a	x	u	f	q	b	c	d	p	f	m	d
u	s	v	y	d	h	i	a	o	z	g	e	u	a	p	b	i	g	d	c
x	b	i	q	u	z	q	v	b	c	k	x	k	x	t	q	q	y	r	q
j	l	d	e	l	c	p	j	j	e	m	p	e	o	h	n	v	k	p	e
b	j	a	k	e	b	k	j	u	q	z	g	u	q	d	p	e	t	l	s
i	t	f	t	j	e	z	n	z	n	n	l	g	k	g	v	k	a	o	y
r	x	s	l	e	w	l	m	y	y	n	f	o	r	j	p	a	a	w	l
k	a	v	g	v	t	z	z	y	o	z	s	q	g	m	r	m	r	n	l
d	m	a	t	i	w	g	l	n	l	e	a	s	x	q	o	v	i	b	d
j	l	f	p	k	k	w	p	e	p	b	v	t	i	r	o	n	r	j	l
m	x	v	x	v	z	i	w	c	z	m	g	o	j	v	z	p	z	k	t
k	e	a	k	p	b	c	l	b	o	s	m	r	j	e	f	x	d	g	n
a	g	h	j	g	c	w	z	j	o	w	r	r	x	i	j	y	v	p	n
y	e	p	f	h	v	c	k	q	l	r	u	g	x	k	d	p	p	b	t
b	p	x	w	i	m	k	o	j	m	a	n	a	u	f	p	l	n	d	z
x	c	f	n	l	l	r	t	a	l	g	o	r	i	t	m	o	s	m	e
s	v	x	m	c	c	u	b	h	l	r	n	b	s	o	z	g	m	n	a
f	q	n	q	r	j	g	e	b	a	s	y	h	l	x	n	b	z	i	j
s	j	t	r	r	t	w	m	z	n	r	w	v	k	b	b	l	i	f	h
b	a	v	d	c	q	t	t	a	r	e	i	x	q	h	k	b	l	i	e

Figura 3: Ejemplo archivo de entrada "sopa.in" con las subcadenas de las palabras marcadas. Nótese que para la palabra "algoritmo" se debería considerar cada substring.

IMPORTANTE

Las letras en la sopa no tendrán tilde, no serán "ñ", y solo se usarán minúsculas.

Salida:

El programa debe generar un archivo de salida de nombre "palabras.out" mostrada en la figura 4, donde se debe indicar la palabra buscada y luego de dos puntos la cantidad de subcadenas encontradas. Luego en cada línea mostrar las subcadenas encontradas ordenadas de menor a mayor según el largo de la subcadena y para palabras del mismo largo el orden es alfabético, esto para cada palabra a buscar.

Laboratorio 2

Todas las coordinaciones

FACULTAD DE INGENIERÍA

Departamento de Ingeniería Informática

Análisis de algoritmos y estructuras de datos



13/05/2020

```
ejemplo:5
eje
emp
jem
plo
jemp
informatica:6
ati
for
mat
nfo
mati
nfor
algoritmo:28
alg
gor
itm
lgo
ori
rit
tmo
algo
gori
itmo
lgor
orit
ritm
algor
gorit
lgori
oritm
ritmo
algori
goritm
lgorit
oritmo
algorit
goritmo
lgoritm
algoritmo
lgoritmo
algoritmo
```

Figura 4: Ejemplo archivo de salida

Laboratorio 2

Todas las coordinaciones

FACULTAD DE INGENIERÍA

Departamento de Ingeniería Informática

Análisis de algoritmos y estructuras de datos



13/05/2020

BONUS OPCIONAL

Implemente la sopa de letras como si fuera una esfera, es decir que desde la posición (0,n-1) se puede buscar en la posición (0,0), lo mismo verticalmente. La realización de este bonus le permitirá subir un punto en cualquier laboratorio entregado.

Según el día que se entregue el laboratorio se evaluará con distinta escala de exigencia.

Fecha de entrega **19/06/2020** hasta las **23:30 hrs** (50% de exigencia).

Fecha de entrega **20/06/2020** hasta las **23:30 hrs** (60% de exigencia).

Fecha de entrega **21/06/2020** hasta las **23:30 hrs** (70% de exigencia).

Recuerde que no entregar alguno de los laboratorios significa la reprobación automática del laboratorio.

Instrucciones de entrega:

- Archivo **PDF** con el **informe** (incluyendo manual de usuario).
- Se debe calcular el **T(n)** y el **O**.
- Se deben graficar los tiempos de ejecución de su programa según distintas entradas y hacer un análisis respecto a los tiempos teóricos.
- Código fuente en archivos **.c** y **.h** (no entregar proyectos de ninguna IDE).
- El código debe permitir ser **compilado en** ambiente **Windows y Linux** por lo que se sugiere usar ANSI C.