seguretat-p1

Marc Sànchez Pifarré

21 de setembre de 2019

```
{r setup, include=FALSE} knitr::opts_chunk$set(echo = TRUE)
```

Informe de la pràctica 1

Apartat a, Cèsar

Què s'ha tingut en compte :

- Donat n com a desplaçament, n >= 0.

Què no s'ha tingut en compte :

- Donat n com a desplaçament, n < 0
- Només es desplacen els caràcters 'a'..'z' alfabet anglés.

Proves

```
entreu un nombre natural corresponent al desplacament: 0
entra el text que vols xifrar: a
TEXT XIFRAT:  a
TEXT ORIGINAL:  a
```

```
entreu un nombre natural corresponent al desplacament: 25
entra el text que vols xifrar: a
TEXT XIFRAT:  z
TEXT ORIGINAL:  a
```

```
entreu un nombre natural corresponent al desplacament: 26
entra el text que vols xifrar: a
TEXT XIFRAT:  a
TEXT ORIGINAL:  a
```

Amb aquestes proves l'algoritme queda testejat i estressat. Suficients per controlar els extrems.

Fem la prova amb un text del lorem Ipsum.

```
entreu un nombre natural corresponent al desplacament: 56253
```

```
entra el text que vols xifrar: Lorem Ipsum is simply dummy text of the printing and typesetting industr
```

```
TEXT XIFRAT:  Ldgtb Iehjb xh hxbean sjbbn itmi du iwt egxcixcv pcs inethtiixcv xcsjhign. Ldgtb Iehjb wp
```

```
TEXT ORIGINAL:  Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum
```

Aquest algoritme permet l'encriptament amb més d'una volta.

Per exemple :

- Encriptem "hola" amb desplaçament = 12 -> taxm.

- Encriptem “taxm” amb desplaçament = 5 -> yfcr
- Desencriptem “yfcr” amb desplaçament = 5 -> taxm
- Desencriptem “taxm” amb desplaçament = 12 -> hola

Apartat b, PolyBios

En aquest apartat hi havia més llibertat a l’hora d’implementar l’algoritme, en el meu cas he optat per fer-lo senzill per poder fer-lo el més genèric possible.

Es representa la taula com un vector v de n posicions on n és el nombre de caràcters de l’alfabet. Per cada posició del vector v s’hi insereix una tupla que conté 3 valors.

- lletra
- caràcter corresponent a la Fila de la taula.
- caràcter corresponent a la columna de la taula.

Definim un text t amb nombre de caràcters m. Llavors la complexitat de l’algoritme és O(n) * m per encriptar i desencriptar.

Propietats negatives :

- Complexitat, es podria haver actuat amb complexitat O(1) i accés directe peró complica el codi i embrut la genericitat a l’hora de generar la taula. (S’ha de controlar si les files son > columnes, si files < columnes o si files == columnes)

Propietats positives :

- Manteniment del codi.
- Escalabilitat en nombre de signes.
- Genericitat en funció del nombre de files i columnes al crear la taula.
- Alteració de la taula de manera senzilla.

S’ha implementat així per simplicitat i per poder adaptar la matriu de les transparències a la pràctica de manera fàcil i fent el menor “marranades” hardcoded possible.

Proves

```
Entreu el nombre de files i columnes, recorda dimensionar correctament la matriu => files >= 5 and colu
Entra el nombre de files : 5
Entra el nombre de columnes : 5
entra el text que vols xifrar: a
TEXT XIFRAT:  AA
TEXT ORIGINAL:  a
```

```
Entreu el nombre de files i columnes, recorda dimensionar correctament la matriu => files >= 5 and colu
Entra el nombre de files : 5
Entra el nombre de columnes : 5
entra el text que vols xifrar: z
TEXT XIFRAT:  EE
TEXT ORIGINAL:  z
```

```
Entreu el nombre de files i columnes, recorda dimensionar correctament la matriu => files >= 5 and colu
Entra el nombre de files : 6
Entra el nombre de columnes : 5
entra el text que vols xifrar: z
TEXT XIFRAT:  FA
TEXT ORIGINAL:  z
```

```
Entreu el nombre de files i columnes, recorda dimensionar correctament la matriu => files >= 5 and colu
Entra el nombre de files : 5
Entra el nombre de columnes : 6
entra el text que vols xifrar: z
TEXT XIFRAT:  EB
TEXT ORIGINAL:  z
```

Fins aquí hem testejat l'algoritme en els seus extrems, ara podem provar alguna possible col · lisió. L'únic cas en que colisionen és amb la i i la j sobre la configuració de 5 files i 5 columnes.

```
Entreu el nombre de files i columnes, recorda dimensionar correctament la matriu => files >= 5 and colu
Entra el nombre de files : 5
Entra el nombre de columnes : 5
entra el text que vols xifrar: i
TEXT XIFRAT:  BD
TEXT ORIGINAL:  i
```

```
Entreu el nombre de files i columnes, recorda dimensionar correctament la matriu => files >= 5 and colu
Entra el nombre de files : 5
Entra el nombre de columnes : 5
entra el text que vols xifrar: j
TEXT XIFRAT:  BD
TEXT ORIGINAL:  i
```

Fem la prova amb un text del lorem Ipsum.

```
Entreu el nombre de files i columnes, recorda dimensionar correctament la matriu => files >= 5 and colu
Entra el nombre de files : 5
Entra el nombre de columnes : 5

entra el text que vols xifrar: Lorem Ipsum is simply dummy text of the printing and typesetting industry

TEXT XIFRAT:  CACDDBAECB BDCEDCDECB BDDC DCBDCBCECAED ADDECBCBED DDAEECDD CDBA DDBCAE CEDBBDCCDDBDCCBB A

TEXT ORIGINAL:  lorem ipsum is simply dummy text of the printing and typesetting industry. lorem ipsum l
```

Aquest algoritme tal i com està muntat no permet més d'una volta d'encriptat.

Apartat c, RailFence

Algoritme per transformació, no cal substituïr caràcters sinó simplement desordenar-los. En aquest cas s'ha optat per actuar amb tots els caràcters sense tenir en copmte si son o no lletres que pertanyen a l'alfabet anglés, es desordena tot!

S'utilitza un vector de list(). Es realitza així degut a que és molt més fàcil de codificar, a l'hora de crear el vector mitjançant l'entrada es fan appends a les llistes de dins del vector. Cada list() simbolitza un rail.

A l'hora de codificar és molt senzill i el mètode té una complexitat de O(n) sent n el nombre de caràcters a processar en funció de l'entrada. (Hi ha un doble bucle peró les iteracions acaben sumant n)

A l'hora de descodificar el nombre de iteracions segueix sent n tot i el doble bucle que hi ha. I és que es genera la taula de raïls a l'inversa de com es va construïr. Per poder fer-ho es requereix el nombre de rails amb el que es va codificar i l'habilitat de veure que n % rails ens donarà el nombre de rails que pot ser que tinguin una lletra de més.

Proves

```
entreu un nombre natural corresponent al Nombre de rails: 1
entra el text que vols xifrar: hola
TEXT XIFRAT: hola
TEXT ORIGINAL: hola
```

```
entreu un nombre natural corresponent al Nombre de rails: 4
entra el text que vols xifrar: hola
TEXT XIFRAT: hola
TEXT ORIGINAL: hola
```

```
entreu un nombre natural corresponent al Nombre de rails: 4
entra el text que vols xifrar: hol
TEXT XIFRAT: hol
TEXT ORIGINAL: hol
```

```
entreu un nombre natural corresponent al Nombre de rails: 4
entra el text que vols xifrar: hola hola
TEXT XIFRAT:  h aohloal
TEXT ORIGINAL:  hola hola
```

Aquí ja tenim testejat l'algoritme en els extrems. Quan hi ha 1 sol carril, quan hi ha tants carrils com caràcters, quan hi ha menys caràcters que carrils i finalment quan hi ha més caràcters que carrils.

```
entreu un nombre natural corresponent al Nombre de rails: 10
entra el text que vols xifrar: Lorem Ipsum is simply dummy text of the printing and typesetting industr

TEXT XIFRAT: Lmyxpntt  i d es rk nea bad u le niiawrhileeimsde usiPivfso  trdirIbnsue ,ui oddksos frael

TEXT ORIGINAL: Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum ha
```

Anàlisi de desencriptació.