## Définitions et notations

On s'intéresse au problème de la recherche d'une chaîne de caractères appelée motif dans une autre chaîne de caractères appelée texte. Plus précisement, on veut lister toutes les occurences (par leur position) du motif dans le texte.
On notera :

## Définitions et notations

On s'intéresse au problème de la recherche d'une chaîne de caractères appelée motif dans une autre chaîne de caractères appelée texte. Plus précisement, on veut lister toutes les occurences (par leur position) du motif dans le texte.
On notera :

- $m$ le motif et $l_m$ sa longueur

## Définitions et notations

On s'intéresse au problème de la recherche d'une chaîne de caractères appelée motif dans une autre chaîne de caractères appelée texte. Plus précisement, on veut lister toutes les occurences (par leur position) du motif dans le texte.
On notera :

- $m$ le motif et $l_m$ sa longueur
- $t$ le texte et $l_t$ sa longueur

## Définitions et notations

On s'intéresse au problème de la recherche d'une chaîne de caractères appelée motif dans une autre chaîne de caractères appelée texte. Plus précisement, on veut lister toutes les occurences (par leur position) du motif dans le texte.
On notera :

- $m$ le motif et $l_m$ sa longueur
- $t$ le texte et $l_t$ sa longueur

D'autre part, parfois le problème se réduira à déterminer si $m$ est présent ou non dans $t$, ou encore on cherchera uniquement la première occurence.

## Définitions et notations

On s'intéresse au problème de la recherche d'une chaîne de caractères appelée motif dans une autre chaîne de caractères appelée texte. Plus précisement, on veut lister toutes les occurences (par leur position) du motif dans le texte.
On notera :

- $m$ le motif et $l_m$ sa longueur
- $t$ le texte et $l_t$ sa longueur

D'autre part, parfois le problème se réduira à déterminer si $m$ est présent ou non dans $t$, ou encore on cherchera uniquement la première occurence.

## Exemple

La recherche du motif $m$=`exe` ($l_m = 3$) dans le texte $t$ =`un excellent exemple et un exercice extraordinaire` ($l_t = 50$) doit produire la liste d'occurrences : [13 ; 27].

## Définitions et notations

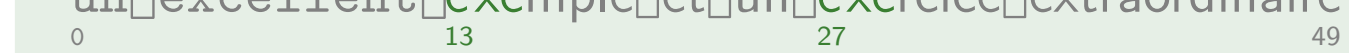
On s'intéresse au problème de la recherche d'une chaîne de caractères appelée motif dans une autre chaîne de caractères appelée texte. Plus précisement, on veut lister toutes les occurences (par leur position) du motif dans le texte.
On notera :

- $m$ le motif et $l_m$ sa longueur
- $t$ le texte et $l_t$ sa longueur

D'autre part, parfois le problème se réduira à déterminer si $m$ est présent ou non dans $t$, ou encore on cherchera uniquement la première occurence.

## Exemple

La recherche du motif $m$=`exe` ($l_m = 3$) dans le texte $t$ =`un excellent exemple et un exercice extraordinaire` ($l_t = 50$) doit produire la liste d'occurrences : [13 ; 27].



## Recherche naïve

Pour recherche si un motif `m` se trouve dans une chaîne `c`, on peut :

1. parcourir chaque caractère de la chaine `c`

## Exemple

Visualisation en ligne du fonctionnement de l'algorithme

## Recherche naïve

Pour recherche si un motif `m` se trouve dans une chaîne `c`, on peut :

1. parcourir chaque caractère de la chaine `c`
2. si ce caractère correspond au premier caractère du motif `m`, alors on avance dans le motif tant que les caractères coïncident.

## Exemple

Visualisation en ligne du fonctionnement de l'algorithme

## Recherche naïve

Pour recherche si un motif `m` se trouve dans une chaîne `c`, on peut :

1. parcourir chaque caractère de la chaine `c`
2. si ce caractère correspond au premier caractère du motif `m`, alors on avance dans le motif tant que les caractères coïncident.
3. si on atteint la fin du motif, alors `m` se trouve dans `c`. Sinon on passe au caractère suivant de `c`.

## Exemple

Visualisation en ligne du fonctionnement de l'algorithme

## Proposition d'implémentation en Python

```
def recherche(motif,chaine):
        lm,lc = len(motif),len(chaine)
        for i in range(lc-lm+1):
                i_motif,i_chaine = 0,i
                while i_motif<lm and chaine[i_chain
                        i_motif += 1
                        i_chaine += 1
                if i_motif == lm:
                        return True
        return False
```

## Coût de la recherche simple

Soient $l_m$ la longueur du motif et $l_c$ celle de la chaine, on vérifie que l'algorithme de recherche simple demande au plus $l_m(l_c - l_m + 1)$ comparaisons

## Exemple

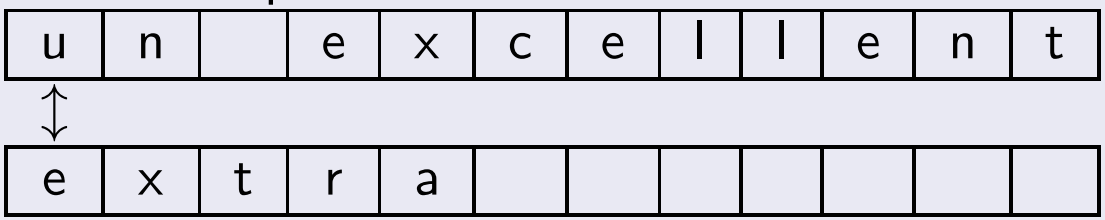
Combien de comparaisons seront nécessaires si on recherche le motif `bbbbbbbbba` (neuf fois le caractère `b` suivi d'un `a`) dans une chaine contenant un million de `b` ?

## Accélération de la recherche

Supposons qu'on recherche le motif `extra` dans la chaine `un excellent exemple et un exercice extraordinaire`. La comparaison naïve ci-dessus commence par :

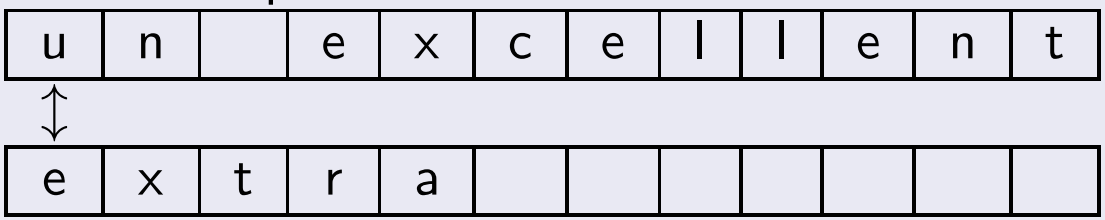
## Accélération de la recherche

Supposons qu'on recherche le motif `extra` dans la chaine `un excellent exemple et un exercice extraordinaire`. La comparaison naïve ci-dessus commence par :



## Accélération de la recherche

Supposons qu'on recherche le motif `extra` dans la chaine `un excellent exemple et un exercice extraordinaire`. La comparaison naïve ci-dessus commence par :




Deux idées vont permettre d'accélérer la recherche :

## Accélération de la recherche

Supposons qu'on recherche le motif `extra` dans la chaine `un excellent exemple et un exercice extraordinaire`. La comparaison naïve ci-dessus commence par :

| u | n | | e | x | c | e | l | l | e | n | t |
|---|---|---|---|---|---|---|---|---|---|---|---|

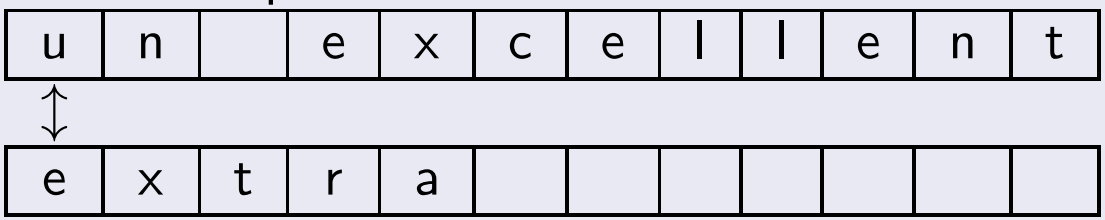
| e | x | t | r | a | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|

Deux idées vont permettre d'accélérer la recherche :

- Commencer par la fin du motif.

## Accélération de la recherche

Supposons qu'on recherche le motif `extra` dans la chaine `un excellent exemple et un exercice extraordinaire`. La comparaison naïve ci-dessus commence par :




Deux idées vont permettre d'accélérer la recherche :

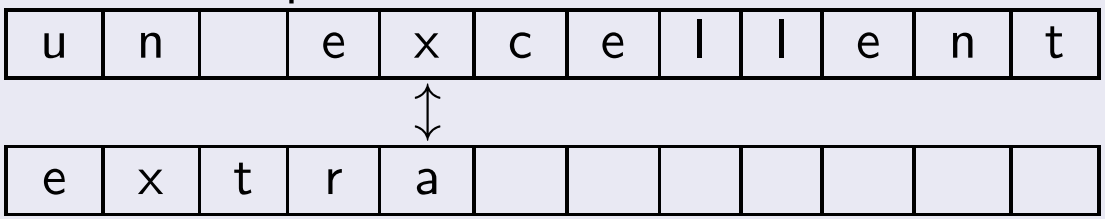
- Commencer par la fin du motif.
- Prétraiter le motif de façon à éviter des comparaisons inutiles.

## Accélération de la recherche

Dans l'exemple ci-dessus cela donne :

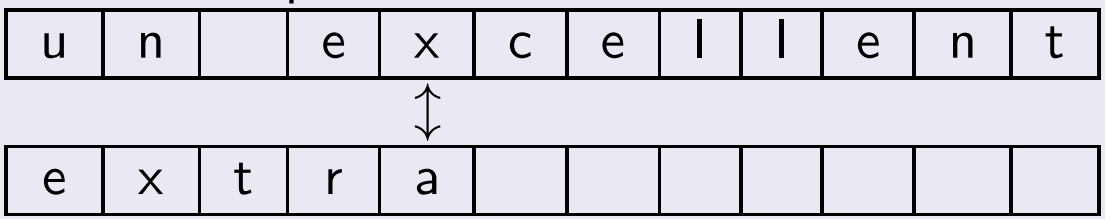
## Accélération de la recherche

Dans l'exemple ci-dessus cela donne :



## Accélération de la recherche

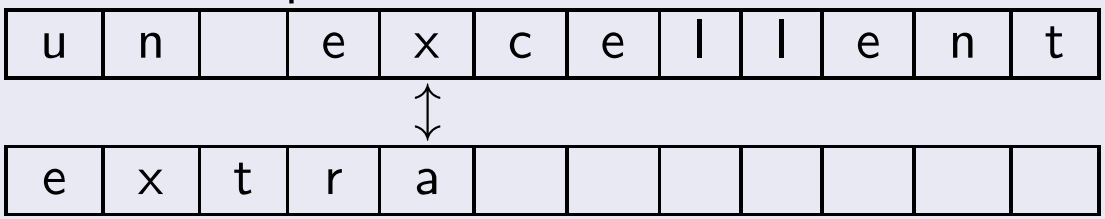
Dans l'exemple ci-dessus cela donne :



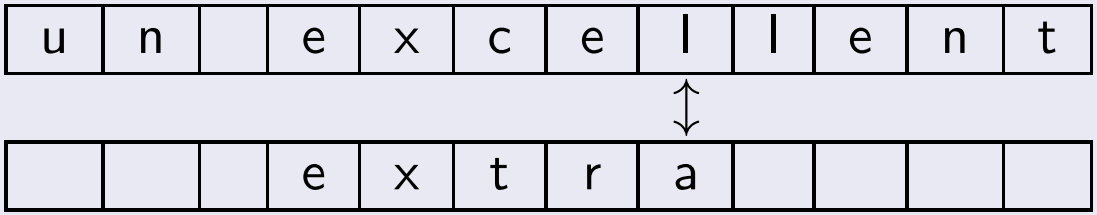

On peut avancer directement de 3 emplacements car le dernier `x` se trouve à 3 emplacements de la fin du motif.

## Accélération de la recherche

Dans l'exemple ci-dessus cela donne :

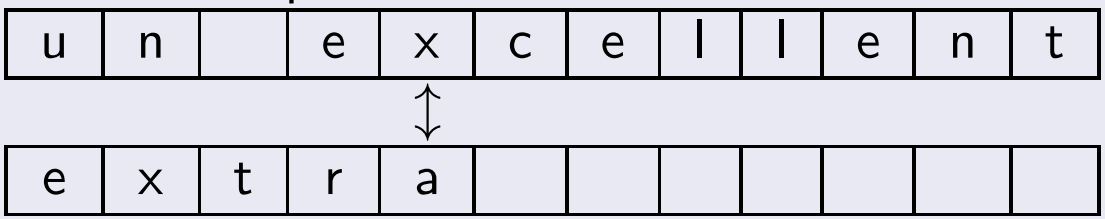



On peut avancer directement de 3 emplacements car le dernier `x` se trouve à 3 emplacements de la fin du motif.

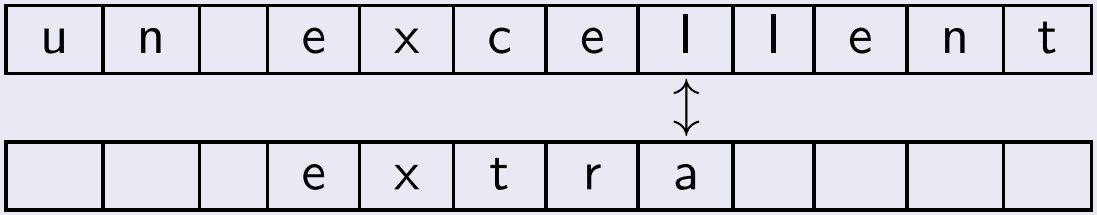


## Accélération de la recherche

Dans l'exemple ci-dessus cela donne :



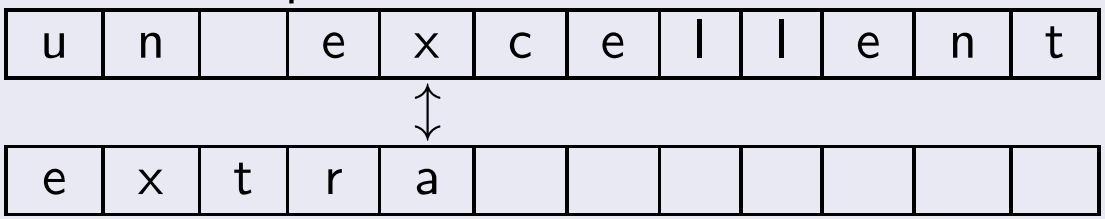

On peut avancer directement de 3 emplacements car le dernier `x` se trouve à 3 emplacements de la fin du motif.



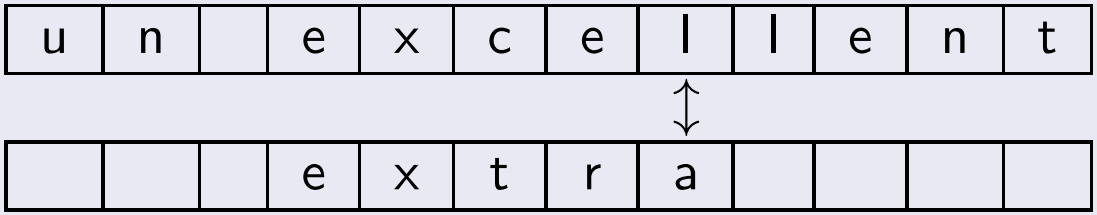

Cette fois, le `l` ne se trouve pas dans le motif, on peut donc avancer de la longueur du motif.

## Accélération de la recherche

Dans l'exemple ci-dessus cela donne :




On peut avancer directement de 3 emplacements car le dernier `x` se trouve à 3 emplacements de la fin du motif.




Cette fois, le `l` ne se trouve pas dans le motif, on peut donc avancer de la longueur du motif.

Visualisation en ligne du fonctionnement de l'algorithme accéléré

## Remarques

- L'implémentation, plus délicate que la recherche naïve fait l'objet d'un exercice.

## Remarques

- L'implémentation, plus délicate que la recherche naïve fait l'objet d'un exercice.
- L'étude du coût de cet algorithme n'est pas au programme, mais à titre d'exemple, on pourra rechercher le nombre de comparaisons de la recherche du motif `aaaaaaaaaa` dans un texte contenant un million de `b` et comparer avec le cas de la recherche naïve.