

# Algorithmique TD11: Algorithme KMP

5 janvier 2010

## 1 Algorithmes

### 1.1 Fonction préfixe d'une chaîne de caractères

```
fonction calculPrefixe(E tableau P(M):caractere ,m:entier):tableau()  
    : caractere  
    variables  
        q,k:entier  
        tableau pi():entier  
    creerTableau(pi,m)  
    pi(1) ← 0  
    k ← 0  
    pour q ← 2 a m pas 1  
        tantque k>0 et P(k+1) ≠ P(q) faire  
            k ← pi(k)  
        ftq  
        si P(k+1)=P(q) alors  
            k ← k+1  
        fsi  
        pi(q) ← k  
    fpour  
    retourner pi  
ffonction
```

### 1.2 Algorithme KMP

```
procedure KMP(E tableau T(N):caractere ,E tableau P(M):caractere ,  
              n:entier ,m:entier)  
    variables  
        tableau pi():caractere  
        i,q:entier  
    pi ← calculPrefixe(P)  
    q ← 0 //nb de caracteres concordant  
    pour i ← 1 a n pas 1 //balaie le texte de gauche a droite  
        tant que q>0 et P(q+1) ≠ T(i) faire  
            q ← pi(q) //prochain caractere ne concorde pas  
        ftq
```

```

si P(q+1)=T(i) alors
  q ← q+1 //prochain caractere concorde
fsi
si q=m alors //est ce que tout P a concordé?
  ecrire("le motif apparait en position"+(i-m))
  q ← pi(q) //chercher prochaine correspondance
fsi
fpour
fprocedure

```

## 2 Exercices

1. Calculer la fonction préfixe pour la chaîne de caractères ababbabbababababbabb.
2. Soit le motif P=bcbcabca et le texte T=abcbcabcbcab
  - (a) Appliquer l'algorithme naïf pour trouver les occurrence de P dans T
  - (b) Appliquer l'algorithme KMP pour trouver les occurrence de P dans T
  - (c) Comparer le nombre de comparaisons nécessaires aux deux algorithmes
3. Etant donné deux chaînes P de taille n et Q de taille m
  - (a) Ecrire un algorithme qui trouve le plus long suffixe de P qui est un préfixe de Q
  - (b) Evaluer sa complexité