

UNIVERSITÉ PIERRE ET MARIE CURIE

ALGORITHMIQUE REPARTIE

---

# Projet Implementation du protocole P2P CHORD

---

*Auteur:*  
WILLIAM FABRE

*Professeur:*  
FRANCK PETIT

Année 2017-2018



## Contents

1	Recherche de cle	2
2	Calcul des finger table	3
3	Insertion d'un pair	4

## Recherche de cle

Creation d'une table avec des elements qui sont unique grace a un algorithme de recherche qui nous evite les doublons. Puis on cree les tables de finger et on les envois. On utilise les tables pour trouver les elements envoyes.

## Calcul des finger table

Pour calculer les finger table il faut d'abord utiliser un algorithme de Hishberg St clair avec l'algorithme d'élection bidirectionnel en  $n \log(n)$  puis 2 parcours de l'anneau soit  $2n$  on est donc sur du  $n$ , un tout petit peu plus fort. A partir de la on a un leader et tout le monde est au courant de son nom. L'elu envoie desormais un message que tout le monde doit signer et lorsqu'il le recupere il applique l'algorithme de l'exercice 1 pour creer toutes les tables et va les diffuser a chaque personne.

## Insertion d'un pair

En faisant l'hypothese qu'on connait quelqu'un a qui envoyer notre message et qu'il va s'occuper de trouver la place de notre nouveau membre. On utilise encore une fois l'exercice 1 et son algo etant donne que nos tables de fingers sont presentes pour trouver la place du nouveau pair sur l'anneau.

Maintenant qu'on a le noeud qui s'occupe de cette clee il faut transformer la clee en node.

Une premiere solution est de refaire l'algorithme de l'exercice 2, une election puis 2 parcours pour prevenir du leader puis recalculer toutes les tables sur un unique node et diffuser.

Une autre serait: On doit deja modifier `Finger[0]` du noeud auquel on appartenait qui devient nous meme. On peut aussi mettre dans notre `finger[0]` l'ancien `finger[0]` de notre ancien node pere. Desormais il faut mettre a jour les nodes qui sont a la plus grande distance possible de nous c'est a dire  $K/2$  gauche et  $K/2$  droite.