

# Algorithmique Appliquée

BTS SIO SISR

Programmes numériques  
simples et techniques de  
débogage

# Plan

- Introduction à la technique "devine-et-vérifie"
- Introduction à la dichotomie
- Introduction à l'instrumentation de code
- Introduction à l'algorithme Newton Raphson
- Histoire des bugs et du debugging dans la culture anglo-saxonne
- Techniques pour déboguer manuellement un programme sur papier
- Utilisation d'un debugger avec points d'arrêt

# Correction du travail à la maison

# DM : Retours sur Scratch et Python

[Lien vers le sujet de DM.](#)

# Introduction à la technique "devine-et-vérifie"

Guess-and-Check

Problème des racines carrées.



# Introduction à la dichotomie

Bisection Search

Le juste prix (faire jouer les étudiants)  
Puis le résoudre  
Puis revenir sur les racines carrées.



# **TD : Utilisation de la dichotomie pour calculer des racines et des logarithmes**

# **TD : Dichotomie pour Racines et Logarithmes**

[Lien vers le sujet de TD.](#)

# Introduction à l'instrumentation de code

Instrumentation à base de `print s`.

Intérêt => comprendre les problèmes.

Instrumentation avec des wallclocks.

Intérêt => amélioration des perfs + comparaison entre solutions

Mentionner rapidement que Python permet de faire de l'instrumentation de code automatique grâce à une notion avancée nommée annotation

# Introduction à l'algorithme Newton Raphson

Présentation d'une meilleure solution pour calculer des racines carrées.



# **TP : Comparaison d'algorithmes ayant le même objectif**



# **TP : Comparaison d'Algorithmes**

[Lien vers le sujet de TP.](#)

# Histoire des bugs et du debugging dans la culture anglo-saxonne

Le mythe de l'insecte dans la machine  
Les vraies origines  
Rappel du Halting Problem et rabachage sur la qualité  
logicielle

# Techniques pour déboguer manuellement un programme sur papier

Tu prends un papier, tu prends un crayon, tu dessines  
un tableau avec tes variables en colonnes et les  
itérations en ligne, pis tu regardes ce qui se passe....  
Et boom => le bug saute aux yeux !

# Utilisation d'un debugger avec points d'arrêt

Démo dans VS Code des breakpoints, breakpoints conditionnels, introspection de variables, etc.



# **TP : Déboguer un programme mal écrit et comportant des bugs**

# **TP : Débogage d'un programme mal écrit**

[Lien vers le sujet de TP.](#)