

Progression de Terminale NSI

- 1. Récursivité** (2 sem)
Fonctions récursives, cas récursif et condition d'arrêt.
Arbre des appels récursifs, déboguer une fonction récursive.
- 2. Programmation Orientée Objet** (2 sem)
Création d'une classe, méthode constructeur, attributs et méthodes.
- 3. Types de données abstraits: Listes, Piles, Files** (2 sem)
Interface vs implémentation.
Listes chaînées implémentées en POO, listes définies récursivement.
Piles et files.
- 4. Base de données** (2 sem)
Modèle relationnel, tables.
Clés primaires, clés étrangères.
Requêtes SQL.
- 5. Types de données abstraits: Arbres** (3 sem)
Généralités sur les arbres : racine, hauteur, profondeur, taille, degré.
Arbre binaire, définition récursive, parcours en profondeur ou en largeur.
Arbre binaire de recherche, insertion d'un élément, parcours d'un arbre, recherche d'un élément.
- 6. Processus** (1 sem)
Notion de processus, état d'un processus, ordonnancement.
Commandes Linux sur les processus.
Interblocage.
- 7. Diviser pour régner** (2 sem)
Diviser, régner, combiner.
Exemples classiques : dichotomie, exponentiation rapide, tri fusion, quart de tour d'une image.
- 8. Routage d'un paquet sur Internet** (2 sem)
Généralités sur le routage, tables de routage.
Routage statique vs routage dynamique. Protocole RIP, protocole OSPF.
- 9. Types de données abstraits: Graphes** (3 sem)
Généralités sur les graphes, matrices d'adjacence et listes d'adjacence.
Algorithmes : parcours en largeur ou en profondeur, recherche d'un cycle, recherche d'un chemin.
- 10. Chiffrement des communications** (1 sem)
Chiffrement symétrique, utilisation du XOR, chiffrement AES.
Chiffrement asymétrique, clés publiques et privées, signature numérique.
Protocole HTTPS.
- 11. Programmation dynamique** (3 sem)
Programmation dynamique descendante récursive avec mémorisation.
Programmation dynamique montante itérative avec remplissage d'un tableau.
Exemples classiques : Fibonacci, rendu de monnaie, sac à dos
- 12. System On Chip** (1 sem)
Loi de Moore
Processeur d'ordinateur de bureau vs SOC de smartphone
Schémas de SOC
- 13. Programmation fonctionnelle** (1 sem)
Paradigme fonctionnel
Fonctions lambda, map, filter, reduce
- 14. Calculabilité, décidabilité** (1 sem)
Programme en tant que donnée
Notion de calculabilité et de décidabilité
Problème de l'arrêt
- 15. Recherche d'un motif dans un texte** (1 sem)
Algorithme de Boyer-Moore-Horspool