

Ce qu'il faut savoir pour l'intra

Section	Liste des points importants pour l'examen
<input type="checkbox"/> Analyse et complexité	- dans le cours on s'est intéressé à la complexité dans le pire des cas
<input type="checkbox"/>	- pour calculer la complexité en temps d'un algorithme, on compte le nombre d' opérations élémentaires qu'on doit exécuter dans le pire des cas
<input type="checkbox"/>	- On utilise la notation asymptotique (grand O) pour exprimer cette complexité
<input type="checkbox"/>	Étant donné des fonctions $f(n)$ et $g(n)$, on dit que $f(n)$ est $O(g(n))$ s'il existe des constantes $c > 0$ et $n_0 \geq 1$ telles que $f(n) \leq c g(n) \quad \forall \text{ entier } n \geq n_0$
<input type="checkbox"/> Méthodes de preuves	- Pas à l'examen
<input type="checkbox"/> Définitions récursives	- savoir définir de façon récursive un ensemble infini d'éléments: 1) commencer par définir le ou les plus petits éléments de l'ensemble 2) expliquer ensuite comment "construire" les autres en nous servant des cas de base
<input type="checkbox"/> Fonctions récursives	- Une fonction récursive est une fonction qui s'appelle elle-même
<input type="checkbox"/>	Les éléments essentiels d'une fonction récursive sont: - un ou plusieurs cas de base - les appels récursifs doivent se terminer sur un cas de base
<input type="checkbox"/>	-Qu'est ce que la récursion linéaire , la récursion binaire ?
<input type="checkbox"/>	- Savoir calculer la complexité en temps d'un algorithme récursif : Nombre d'appels récursifs X complexité du cas de base
<input type="checkbox"/> Types abstraits	- Qu'est qu'un type abstrait de données ?
<input type="checkbox"/> Piles	- "Dernier arrivé, premier servi"
<input type="checkbox"/>	- Opérations principales : 1) empiler(objet) 2) dépiler()
<input type="checkbox"/>	- Exception PileVide si on essaie de dépiler une pile vide
<input type="checkbox"/>	- Implémentations : 1) avec un tableau : - On doit prédéfinir une taille N - On a une exception PilePleine liée à l'implémentation - On doit être capable de définir en pseudocode les opérations principales - Complexité en espace: $O(N)$ - Complexité en temps de dépiler() et empiler(objet): $O(1)$
<input type="checkbox"/>	2) avec une liste chaînée - Pas de nouvelle exception liée à l'implémentation - On doit être capable de définir en pseudocode les opérations principales - Complexité en espace: $O(n)$ si n objets empilés - Complexité en temps de dépiler() et empiler(objet): $O(1)$
<input type="checkbox"/> Files	- "Premier arrivé, premier servi"
<input type="checkbox"/>	- Opérations principales : 1) ajouter(objet) 2) enlever()
<input type="checkbox"/>	- Exception FileVide si on essaie d'enlever un élément d'une file vide

	Section	Liste des points importants pour l'examen
<input type="checkbox"/>		<p>- Implémentations: 1) avec un tableau circulaire :</p> <ul style="list-style-type: none"> - On doit prédéfinir une taille N - On a une exception FilePleine liée à l'implémentation - On doit être capable de définir en pseudocode les opérations principales - Complexité en espace: $O(N)$ - Complexité en temps de dépiler() et empiler(objet): $O(1)$
<input type="checkbox"/>		<p>2) avec une liste chaînée</p> <ul style="list-style-type: none"> - Pas de nouvelle exception liée à l'implémentation - On doit être capable de définir en pseudocode les opérations principales - Complexité en espace: $O(n)$ si n objets dans la file - Complexité en temps de enlever() et ajouter(objet): $O(1)$
<input type="checkbox"/>	Arbres, arbres binaires	- Se rappeler des définitions et de la terminologie
<input type="checkbox"/>		- Profondeur d'un noeud et hauteur d'un arbre
<input type="checkbox"/>		<p>- Les différents parcours d'arbres:</p> <ul style="list-style-type: none"> - En profondeur: 1) Parcours Préfixe (preorder) 2) Parcours Suffixe (postorder) 3) Parcours Symétrique (inorder) - En largeur: 1) Parcours hiérarchique
<input type="checkbox"/>		- TAD arbres binaires + définition
<input type="checkbox"/>		- Propriété de hauteur des arbres binaires
<input type="checkbox"/>		- Code de Huffman
<input type="checkbox"/>		Implémentations: 1) table des parents: avantages, inconvénients
<input type="checkbox"/>		2) tableau: comment ça marche? Avantages, inconvénients
<input type="checkbox"/>	Files avec priorité	- définition + TAD files avec priorités
<input type="checkbox"/>		- Opérations principales: 1) insérer(k,v) 2) enleverMin()
<input type="checkbox"/>		- relation d' ordre total sur les clés
<input type="checkbox"/>		- comparateur: à quoi ça sert? Pseudocode dans un cas particulier
<input type="checkbox"/>		<p>- Trier avec une file avec priorités:</p> <ol style="list-style-type: none"> 1) On insère les éléments à trier un à un dans la file 2) On tri avec une suite d'opérations enleverMin()
<input type="checkbox"/>		<p>Implémentations: 1) séquence ordonnée:</p> <ul style="list-style-type: none"> - insérer(k,v): $O(n)$ - enleverMin: $O(1)$ - trier: $O(n^2)$
<input type="checkbox"/>		<p>2) séquence non-ordonnée:</p> <ul style="list-style-type: none"> - insérer(k,v): $O(1)$ - enleverMin: $O(n)$ - trier: $O(n^2)$
<input type="checkbox"/>		<p>3) Monceau:</p> <ul style="list-style-type: none"> - insérer(k,v): $O(\log n)$ - enleverMin: $O(\log n)$ - trier: $O(n \log n)$
<input type="checkbox"/>	Monceaux	Définition, hauteur, comment insérer, comment supprimer, implémentation