Rattrapage 2022-2023 - CYBER1 (1h30)

Algo et Structure de Données 2

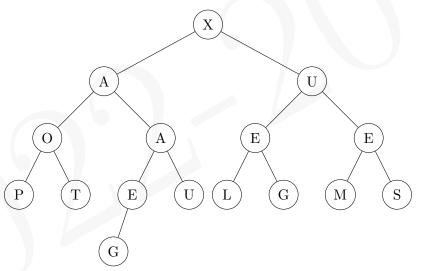
	,
NOM:	PRÉNOM :

Vous devez respecter les consignes suivantes, sous peine de 0 :

- I) Lisez le sujet en entier avec attention
- II) Répondez sur le sujet
- III) Ne détachez pas les agrafes du sujet
- IV) Écrivez lisiblement vos réponses (si nécessaire en majuscules)
- V) Vous devez écrire dans le langage algorithmique classique ou en C (donc pas de Python ou autre)
- VI) Ne trichez pas

1 Arbres Binaires (10 points)

1.1 (3 points) Indiquez toutes les propriétés que possède cet arbre, puis écrivez les clés lors d'un parcours profondeur main gauche de l'arbre dans les 3 ordres ainsi que lors d'un parcours largeur :



Arité :			Tail	le:		Н	lauteu	ır :		Nb:	feuille	s:			
Arbre binain Arbre binain Peigne gauce	re parfa	,	calen	nent c	omple	et		Arbre	binai filifor	rme	resque	e) con	nplet		
Parcours profe	ondeur	:													
ordre préfixe :	-	-	_	-	-	_	-	-	-	-	-	-	_	-	-
ordre infixe :	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ordre suffixe :	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Parcours large	eur:														
ordre:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

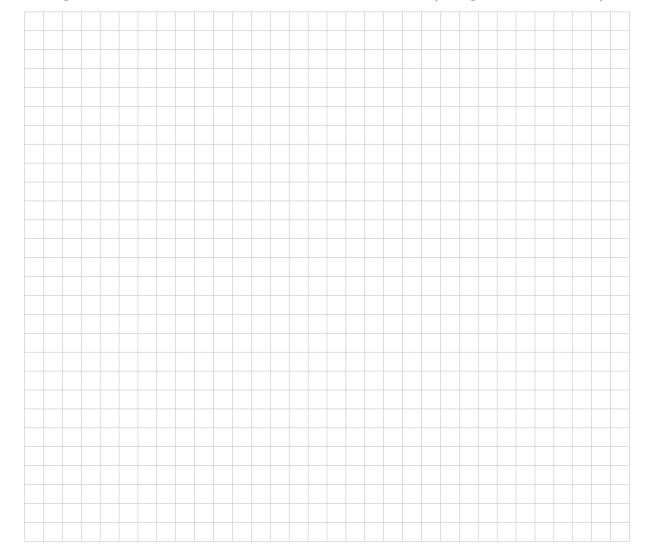
1.2 (4 points) Dessinez le résultat de l'insertion dans cet ordre précis des éléments suivants dans un ABR (insertion en feuille) et dans un AVL :

Éléments insérés : 32 - 8 - 16 - 96 - 64 - 72 - 24 - 4

ABR AVL

1.3 (3 points) Écrivez une fonction récursive « parc_prof_rec » effectuant un parcours profondeur main gauche dans un arbre binaire, et affichant les nœuds dans chacun des ordres :

Il faut expliciter les éventuels ordres au format : « Ordre : nœud » (exemple : « Préfixe : 42 »)



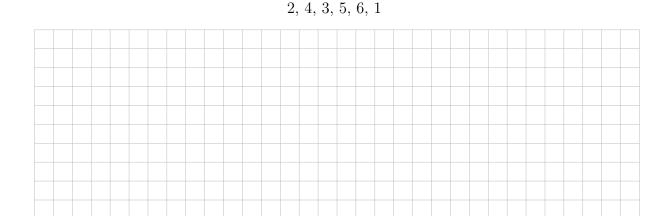
2 Arbres Binaires: Parcours Largeur (10 points)

2.1 (2 points) Effectuez les opérations suivantes et affichez la structure dans son état final :

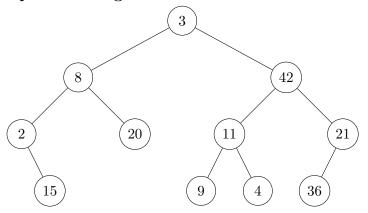
		Pile		File			
ii iii iv v vi vii	push 5 push 12 push 14 pop push 7 push 6 pop pop	ix pop x push 2 xi push 8 xii push 11 xiii pop	i enqueue 5 ii enqueue 12 iii enqueue 14 iv dequeue v enqueue 7 vi enqueue 6 vii dequeue viii dequeue	ix dequeue x enqueue 2 xi enqueue 5 xii enqueue 11 xiii dequeue			

- 1) Quelle est la spécificité d'une pile concernant l'ordre d'entrée et de sortie des éléments?
- 2) Quelle est la spécificité d'une file concernant l'ordre d'entrée et de sortie des éléments?
- 2.2 (1 point) En admettant que l'on dispose d'une pile et que l'on insère les données « 1 2 3 4 5 6 » dans cet ordre exclusivement, décrivez les scénarios permettant d'obtenir les sorties suivantes :

exemple : pour « A B C » en entrée, on peut obtenir « B C A » en sortie en faisant : « push A », « push B », « pop », « push C », « pop », « pop »
On a bien inséré A, puis B, puis C, mais l'ordre de sortie est différent suivant les « pop »



2.3 (2 points) À partir de l'arbre affiché, répondez aux questions et effectuez le parcours largeur :



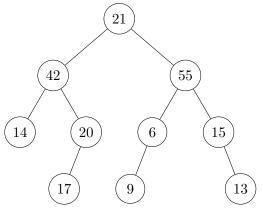
1) Quelle structure est requise pour effectuer un parcours largeur?

2) Effectuez le parcours largeur de l'arbre en détaillant pas à pas l'état de la structure associée.

Structure : Nœud traité : \emptyset 3 \emptyset 3

Structure : Nœud traité :

2.4~ (1 point) À partir du tableau et de l'arbre affiché, répondez à la question suivante :

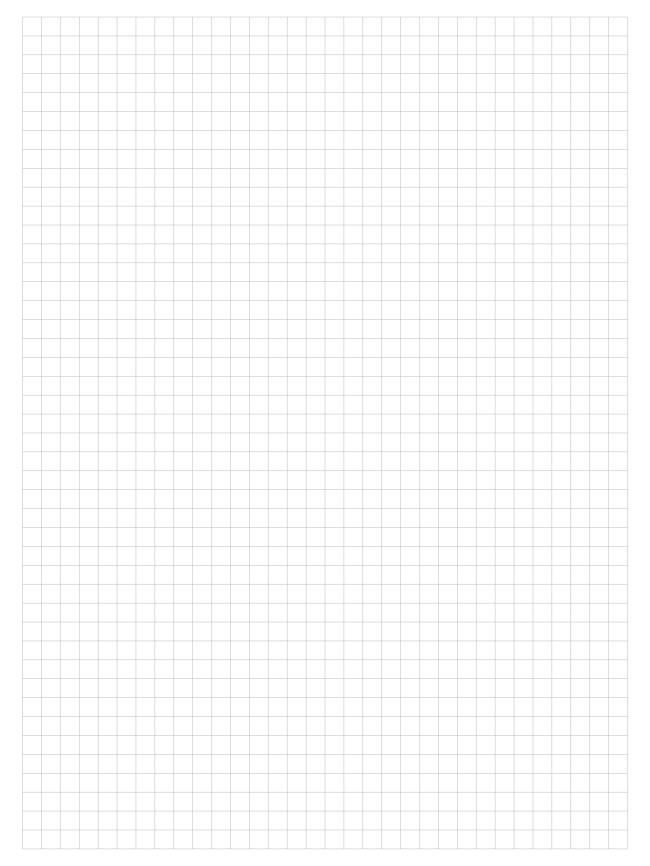


15	6	20	14

Dans le cas d'un parcours largeur, quel nœud est actuellement traité d'après l'état de la structure?

2.5 (4 points) Écrivez une fonction itérative « $parc_larg$ » effectuant un parcours largeur dans un arbre binaire, et affichant chacun des nœuds dans l'ordre hiérarchique :

Vous pouvez utiliser les structures externes : stack_t (create, push, head, pop, delete) queue_t (create, enqueue, head, dequeue, delete)



RATTRAPAGE ALGORITHMIQUE ET STRUCTURES DE DONNÉES 2