Objectif

- Complexité en moyenne.

Problème [Complexité en moyenne du tri rapide]

1. Réaliser un suivi à la trace de la procédure `partitionBis` appliquée aux instances suivantes :

$$I_1 = [2,6,0,4,3,1,5], I_2 = [7,6,5,4,3,2,1] \text{ et } I_3 = [1,2,3,4,5,6,7].$$

Instruction	Description/Remarque	T	indpiv	pospiv	x	j
partitionBis(I_1, 0, 5)	T <- I_1, deb <- 0, fin <- 5	[2, 6, 0, 4, 3, 1, 5]				
indpiv <- deb	indipiv <- 0	[2, 6, 0, 4, 3, 1, 5]	0			
pospiv <- deb	pospiv <- 0	[2, 6, 0, 4, 3, 1, 5]	0	0		
x <- T(deb)	x <- T(0)	[2, 6, 0, 4, 3, 1, 5]	0	0	2	
[Pour] j <- deb + 1	j <- 0 + 1	[2, 6, 0, 4, 3, 1, 5]	0	0	2	1
[Pour] j <= fin	1 < 5 est vrai, la boucle continue	[2, 6, 0, 4, 3, 1, 5]	0	0	2	1
[Si] T(j) <= x	6 <= 2 est faux, on retourne a la boucle	[2, 6, 0, 4, 3, 1, 5]	0	0	2	1
[Pour] j <- j + 1	j < - 1 + 1	[2, 6, 0, 4, 3, 1, 5]	0	0	2	2
[Pour] j <= fin	2 <= 5 est vrai, la boucle continue	[2, 6, 0, 4, 3, 1, 5]	0	0	2	2
[Si] T(j) <= x	0 <= 2 est vrai	[2, 6, 0, 4, 3, 1, 5]	0	0	2	2
pospiv <- pospiv + 1	pospiv <- 0 + 1	[2, 6, 0, 4, 3, 1, 5]	0	1	2	2
[Si] j >pospiv	2 > 1 est vrai	[2, 6, 0, 4, 3, 1, 5]	0	1	2	2
echanger(T, pospiv, j)	T(1) <- 0, T(2) <- 6	[2, 0, 6, 4, 3, 1, 5]	0	1	2	2
[Pour] j <- j + 1	j < - 2 + 1	[2, 0, 6, 4, 3, 1, 5]	0	1	2	3
[Pour] j <= fin	3 <= 5 est vrai, la boucle continue	[2, 0, 6, 4, 3, 1, 5]	0	1	2	3
[Si] T(j) <= x	4 <= 2 est faux	[2, 0, 6, 4, 3, 1, 5]	0	1	2	3
[Pour] j <- j + 1	j < - 3 + 1	[2, 0, 6, 4, 3, 1, 5]	0	1	2	4
[Pour] j <= fin	4 <= 5 est vrai, la boucle continue	[2, 0, 6, 4, 3, 1, 5]	0	1	2	4
[Si] T(j) <= x	3 <= 2 est faux	[2, 0, 6, 4, 3, 1, 5]	0	1	2	4
[Pour] j <- j + 1	j < - 4 + 1	[2, 0, 6, 4, 3, 1, 5]	0	1	2	5
[Pour] j <= fin	5 <= 5 est vrai, la boucle continue	[2, 0, 6, 4, 3, 1, 5]	0	1	2	5
[Si] T(j) <= x	1 <= 2 est vrai	[2, 0, 6, 4, 3, 1, 5]	0	1	2	5
pospiv <- pospiv + 1	pospiv <- 1 + 1	[2, 0, 6, 4, 3, 1, 5]	0	2	2	5
[Si] j >pospiv	5 > 2 est vrai	[2, 0, 6, 4, 3, 1, 5]	0	2	2	5
echanger(T, pospiv, j)	T(2) <- 1, T(5) <- 6	[2, 0, 1, 4, 3, 6, 5]	0	2	2	5
[Pour] j <- j + 1	j < - 5 + 1	[2, 0, 1, 4, 3, 6, 5]	0	2	2	6
[Pour] j <= fin	6 <= 5 est faux, on sort de la boucle	[2, 0, 1, 4, 3, 6, 5]	0	2	2	6
[Si] indpiv < pospiv	0 < 2 est vrai	[2, 0, 1, 4, 3, 6, 5]	0	2	2	6
echanger(T, indpiv, pospiv)	T(0) <- 1, T(2) <- 2	[1, 0, 2, 4, 3, 6, 5]	0	2	2	6

Nombre d'**échanges** : 3, Nombre de **comparaison** : 5

Notre tableau en sorti $[1,0,2,4,3,6,4]$ a été partitionné en deux parties autour du pivot 2. A gauche, l'on a $T_g = [1,0]$ dont tous les éléments $g \in T_g$ satisfaitent $g < 2$ et à droite l'on a $T_d = [4,3,6,5] \mid \forall d \in T_d,\ d > 2$. La schéma de partition de l'algorithme `partitionBis` parcours le tableau pour compter le nombre des valeurs qui sont inférieures au pivot. En le comptant, s'il il y a des valuers plus grand que le pivot dans le soustableau à gauche, `partitionBis` va effectuer un échange. Donc, on verra que ce schéma doit opérer $n-2$ comparaisons des éléments du tableau à trier.

Instruction	Description/Remarque	T	indpiv	pospiv	x	j
partitionBis(I_2, 0, 5)	T <- I_2, deb <- 0, fin <- 5	[7, 6, 5, 4, 3, 2, 1]				
indpiv <- deb	indpiv <- 0	[7, 6, 5, 4, 3, 2, 1]	0			
pospiv <- deb	pospiv <- 0	[7, 6, 5, 4, 3, 2, 1]	0	0		
x <- T(deb)	x <- 7	[7, 6, 5, 4, 3, 2, 1]	0	0	7	
[Pour] j <- deb + 1	j <- 0 + 1	[7, 6, 5, 4, 3, 2, 1]	0	0	7	1
[Pour] j <= fin	1 <= 5 est vrai, la boucle continue	[7, 6, 5, 4, 3, 2, 1]	0	0	7	1
[Si] T(j) <= x	6 <= 7 est vrai	[7, 6, 5, 4, 3, 2, 1]	0	0	7	1
pospiv <- pospiv + 1	pospiv <- 0 + 1	[7, 6, 5, 4, 3, 2, 1]	0	1	7	1
[Si] j > pospiv	1 > 1 est faux	[7, 6, 5, 4, 3, 2, 1]	0	1	7	1
[Pour] j <- j + 1	j <- 1 + 1	[7, 6, 5, 4, 3, 2, 1]	0	1	7	2
[Pour] j <= fin	2 <= 5 est vrai, la boucle continue	[7, 6, 5, 4, 3, 2, 1]	0	1	7	2
[Si] T(j) <= x	5 <= 7 est vrai	[7, 6, 5, 4, 3, 2, 1]	0	1	7	2
pospiv <- pospiv + 1	pospiv <- 1 + 1	[7, 6, 5, 4, 3, 2, 1]	0	2	7	2
[Si] j > pospiv	2 > 2 est faux	[7, 6, 5, 4, 3, 2, 1]	0	2	7	2
[Pour] j <- j + 1	j <- 2 + 1	[7, 6, 5, 4, 3, 2, 1]	0	2	7	3
[Pour] j <= fin	3 <= 5 est vrai, la boucle continue	[7, 6, 5, 4, 3, 2, 1]	0	2	7	3
[Si] T(j) <= x	4 <= 7 est vrai	[7, 6, 5, 4, 3, 2, 1]	0	2	7	3
pospiv <- pospiv + 1	pospiv <- 2 + 1	[7, 6, 5, 4, 3, 2, 1]	0	3	7	3
[Si] j > pospiv	3 > 3 est faux	[7, 6, 5, 4, 3, 2, 1]	0	3	7	3
[Pour] j <- j + 1	j <- 3 + 1	[7, 6, 5, 4, 3, 2, 1]	0	3	7	4
[Pour] j <= fin	4 <= 5 est vrai, la boucle continue	[7, 6, 5, 4, 3, 2, 1]	0	3	7	4
[Si] T(j) <= x	3 <= 7 est vrai	[7, 6, 5, 4, 3, 2, 1]	0	3	7	4
pospiv <- pospiv + 1	pospiv <- 3 + 1	[7, 6, 5, 4, 3, 2, 1]	0	4	7	4
[Si] j > pospiv	4 > 4 est faux	[7, 6, 5, 4, 3, 2, 1]	0	4	7	4
[Pour] j <- j + 1	j <- 4 + 1	[7, 6, 5, 4, 3, 2, 1]	0	4	7	5
[Pour] j <= fin	5 <= 5 est vrai, la boucle continue	[7, 6, 5, 4, 3, 2, 1]	0	4	7	5
[Si] T(j) <= x	2 <= 7 est vrai	[7, 6, 5, 4, 3, 2, 1]	0	4	7	5
pospiv <- pospiv + 1	pospiv <- 4 + 1	[7, 6, 5, 4, 3, 2, 1]	0	5	7	5
[Si] j > pospiv	5 > 5 est faux	[7, 6, 5, 4, 3, 2, 1]	0	5	7	5
[Pour] j <- j + 1	j <- 5 + 1	[7, 6, 5, 4, 3, 2, 1]	0	5	7	6
[Pour] j <= fin	6 <= 5 est faux, la boucle termine	[7, 6, 5, 4, 3, 2, 1]	0	5	7	6
[Si] indpiv < pospiv	0 < 5 est vrai	[7, 6, 5, 4, 3, 2, 1]	0	5	7	6
echanger(T, indpiv, pospiv)	T(0) <- 1, T(5) <- 7	[1, 6, 5, 4, 3, 2, 7]	0	5	7	6

Nombre d'**échanges** : 1, Nombre de **comparaison** : 5

Chaque élément du tableau qu'on à comparé avec le pivot en était inférieur. Cependant, on a réussi a partitionner le tableau avec un seul échange dans le dernier étape. Avec le pivot 7, $T_g = [1,6,5,4,3,2]$ et il vérifie $g < 7 \ \forall g \in T_g$.

Instruction	Description/Remarque	T	indpiv	pospiv	x	j
partitionBis(I_3, 0, 5)	T <- I_3, deb <- 0, fin <- 5	[1, 2, 3, 4, 5, 6, 7]				
indpiv <- deb	indpiv <- 0	[1, 2, 3, 4, 5, 6, 7]	0			
pospiv <- deb	pospiv <- 0	[1, 2, 3, 4, 5, 6, 7]	0	0		
x <- T(deb)	x <- 1	[1, 2, 3, 4, 5, 6, 7]	0	0	1	
[Pour] j <- deb + 1	j <- 0 + 1	[1, 2, 3, 4, 5, 6, 7]	0	0	1	1
[Pour] j <= fin	1 <= 5 est vrai, la boucle continue	[1, 2, 3, 4, 5, 6, 7]	0	0	1	1
[Si] T(j) <= x	2 <= 1 est faux	[1, 2, 3, 4, 5, 6, 7]	0	0	1	1
[Pour] j <- j + 1	j <- 1 + 1	[1, 2, 3, 4, 5, 6, 7]	0	0	1	2
[Pour] j <= fin	2 <= 5 est vrai, la boucle continue	[1, 2, 3, 4, 5, 6, 7]	0	0	1	2
[Si] T(j) <= x	3 <= 1 est faux	[1, 2, 3, 4, 5, 6, 7]	0	1	1	2
[Pour] j <- j + 1	j <- 2 + 1	[1, 2, 3, 4, 5, 6, 7]	0	0	1	3
[Pour] j <= fin	3 <= 5 est vrai, la boucle continue	[1, 2, 3, 4, 5, 6, 7]	0	0	1	3
[Si] T(j) <= x	4 <= 1 est faux	[1, 2, 3, 4, 5, 6, 7]	0	1	1	3
[Pour] j <- j + 1	j <- 3 + 1	[1, 2, 3, 4, 5, 6, 7]	0	0	1	4
[Pour] j <= fin	4 <= 5 est vrai, la boucle continue	[1, 2, 3, 4, 5, 6, 7]	0	0	1	4
[Si] T(j) <= x	5 <= 1 est faux	[1, 2, 3, 4, 5, 6, 7]	0	1	1	4
[Pour] j <- j + 1	j <- 4 + 1	[1, 2, 3, 4, 5, 6, 7]	0	0	1	5
[Pour] j <= fin	5 <= 5 est vrai, la boucle continue	[1, 2, 3, 4, 5, 6, 7]	0	0	1	5
[Si] T(j) <= x	6 <= 1 est faux	[1, 2, 3, 4, 5, 6, 7]	0	1	1	5
[Pour] j <- j + 1	j <- 5 + 1	[1, 2, 3, 4, 5, 6, 7]	0	0	1	6
[Pour] j <= fin	6 <= 5 est faux, la boucle termine	[1, 2, 3, 4, 5, 6, 7]	0	0	1	6
[Si] indpiv < pospiv	0 < 0 est faux	[1, 2, 3, 4, 5, 6, 7]	0	0	1	6

Nombre d'**échanges** : 0, Nombre de **comparaison** : 5

Tout les éléments de $I_3 = [1,2,3,4,5,6,7]$ sont déjà triés donc on effectue aucun échange. On constate qu'il y a encore $n-2=5$ comparaisons parce qu'on a traversé le tableau à partir du premier élément jusqu'a l'avant-dernier élément. Le tableau est, bien entendu, bien partitionné autor $p = \mathbf{1}$, tel que $T_d = [2,3,4,5,6,7]$.

2. Démontrer que la procédure `partitionBis` est correcte et analyser sa complexité.

Pour démontrer que la procédure `partitionBis` est correcte on doit prouver que donné n'importe quel instance, l'algorithme `partitionBis` renvoie un tableau qui est partitionné. Pour procéder, on définit donc un tableau partitionné est celui dont tous les éléments à gauche d'un élément dit pivot est inférieure au pivot, et tout élément à droite du pivot est supérieure au pivot.

3. Quels changements, s'ils existent, à apporter au pseudo-code du tri rapide ?
4. Conduire une analyse de complexité en moyenne du tri rapide utilisant la procédure `partitionBis` à la place de la procédure `partition`.
5. D'après votre expérimentation, laquelle des deux méthodes `partition` et `partitionBis` est la plus efficace ?