

### Algorithme d'ajout :

2 déplacements maximum seront nécessaires pour ajouter un écolier tout en maintenant le tri de la table !

Cas 1 : l'écolier à ajouter est rouge.  
Celui-ci est placé après le dernier rouge.

Cas 2 : l'écolier à ajouter est jaune.  
Le rouge qui se trouve après le dernier jaune est placé après le dernier rouge.  
Il laisse donc une place libre après le dernier jaune.  
Le nouvel écolier est placé à cet endroit qui vient de se libérer.

Cas 3 : l'écolier à ajouter est noir.  
Le rouge qui est placé après le dernier jaune est placé en fin de table.  
Il laisse donc une place libre après le dernier jaune.  
Le jaune qui est placé après le dernier noir est placé à cet endroit qui vient de se libérer.  
Il laisse donc une place libre après le dernier noir.  
Le nouvel écolier est placé à cet endroit qui vient de se libérer.

### Exemples :

Cas 1 :

tableTriée avant :

nora	nico	noel	nestor	julie	jo	rene	remi	robin	
------	------	------	--------	-------	----	------	------	-------	--

Aucun déplacement n'est nécessaire.

tableTriée après ajout de **ralph** :

nora	nico	noel	nestor	julie	jo	rene	remi	robin	<b>ralph</b>
------	------	------	--------	-------	----	------	------	-------	--------------

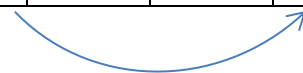
Cas 2 :

tableTriée avant :

nora	nico	noel	nestor	julie	jo	rene	remi	robin	
------	------	------	--------	-------	----	------	------	-------	--

Pour ajouter **jean**, 1 déplacement est nécessaire :

nora	nico	noel	nestor	julie	jo	rene	remi	robin	
------	------	------	--------	-------	----	------	------	-------	--



tableTriée après ajout de **jean** :

nora	nico	noel	nestor	julie	jo	<b>jean</b>	remi	robin	<b>rene</b>
------	------	------	--------	-------	----	-------------	------	-------	-------------

Cas 3 :

tableTrie avant :

nora	nico	noel	nestor	julie	jo	rene	remi	robin	
------	------	------	--------	-------	----	------	------	-------	--

Pour ajouter nick, 2 déplacements sont nécessaires :

nora	nico	noel	nestor	julie	jo	rene	remi	robin	
					(2)		(1)		

tableTrie après ajout de nick :

nora	nico	noel	nestor	nick	jo	julie	remi	robin	rene
------	------	------	--------	------	----	-------	------	-------	------

### Algorithme de suppression :

3 déplacements maximum seront nécessaires pour supprimer un écolier tout en maintenant le tri de la table et en ne laissant pas de « trou » dans la table.

Cas 1 : l'écolier à supprimer est rouge.

Celui-ci est remplacé par le dernier rouge.

Cas 2 : l'écolier à supprimer est jaune.

Celui-ci est remplacé par le dernier jaune.

Le dernier jaune est remplacé par le dernier rouge.

Cas 3 : l'écolier à supprimer est noir.

Celui-ci est remplacé par le dernier noir.

Le dernier noir est remplacé après le dernier jaune.

Le dernier jaune est remplacé par le dernier rouge.

### Exemples :

Cas 1 :

tableTrie avant :

nora	nico	noel	nestor	julie	jo	rene	remi	robin	
------	------	------	--------	-------	----	------	------	-------	--

Pour supprimer **rene**, 1 déplacement est nécessaire :

nora	nico	noel	nestor	julie	jo	rene	remi	robin	

tableTrie après suppression de **rene** :

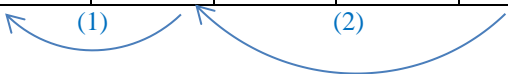
nora	nico	noel	nestor	julie	jo	robin	remi		
------	------	------	--------	-------	----	-------	------	--	--

Cas 2 :

tableTrie avant :

nora	nico	noel	nestor	julie	jo	rene	remi	robin	
------	------	------	--------	-------	----	------	------	-------	--

Pour supprimer **julie**, 2 déplacements sont nécessaires :

nora	nico	noel	nestor	julie	jo	rene	remi	robin	
									

tableTrie après suppression de **julie** :

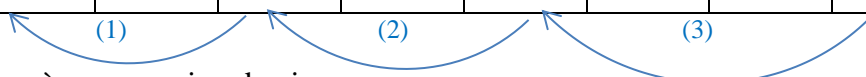
nora	nico	noel	nestor	jo	robin	rene	remi		
------	------	------	--------	----	-------	------	------	--	--

Cas 3 :

tableTrie avant :

nora	nico	noel	nestor	julie	jo	rene	remi	robin	
------	------	------	--------	-------	----	------	------	-------	--

Pour supprimer nico, 3 déplacements sont nécessaires :

nora	nico	noel	nestor	julie	jo	rene	remi	robin	
									

tableTrie après suppression de nico :

nora	nestor	noel	jo	julie	robin	rene	remi		
------	--------	------	----	-------	-------	------	------	--	--