

Analyse et mise en œuvre de tris de tables feuille réponse



Nom :	Note :
	Classe :

1 Le tri par sélection

Q1. Faire fonctionner la suite du tri à la main, la première étape a déjà été réalisée, compléter le tableau sur la feuille réponse :

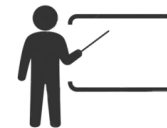
10	56	-2	52	-8	41	13
-8	56	-2	52	10	41	13



Script1. Coder l'algorithme de tri par sélection en Python [tri par selection depart eleve 2024.py](#)



Faites fonctionner votre algorithme et remplir le tableau suivant .



Tri par sélection en S

n valeurs	5000	10000	15000	20000	25000	30000
Durée						

Q2. En exploitant vos résultats précédents, donner une indication sur la complexité de l'algorithme de tri par sélection.

2 Le tri par insertion



Compléter le script Python pour réaliser un tri par insertion d'une liste. [tri par insertion depart eleve 2024.py](#)



Q3. Vérifier expérimentalement que le comportement de l'algorithme dans le cas défavorable est bien d'une complexité en $O(n^2)$.

Tri par insertion cas défavorable en S

n valeurs	5000	10000	15000	20000	25000	30000
Durée						

3 Un autre exemple de tri le tri à bulles

Q4. En observant la table ci-dessous que l'on veut trier avec les valeurs les plus élevées à droite. Indiquer combien d'échanges vont être réalisés sur la première passe de l'algorithme.

-10	24	-2	-12	30	41	13
-----	----	----	-----	----	----	----

Q5. Même question pour celle-ci ?

-4	0	1	12	21	34	43
----	---	---	----	----	----	----

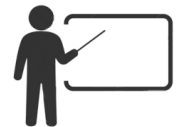
Q6. Faire fonctionner à la main : compléter le tableau ci-dessous :

10	56	-2	52	-8	41	13
10	-2	52	-8	41	13	56



Script4. Compléter l'algorithme du tri à bulles en Python.

[tri a bulles depart eleve 2024.py](#)



Q7. Vérifier expérimentalement que le comportement de l'algorithme est bien d'une complexité en $O(n^2)$.

Tri à bulles en S

n valeurs	5000	10000	15000	20000	25000	30000
Durée						