Algos de tri

**I. Tri par insertion**

* **Principe :**  
  Parcourt le tableau et construit progressivement une partie triée. À chaque itération, l’élément suivant est inséré à la bonne position dans la partie déjà triée, en décalant les éléments plus grands vers la droite.

**II. Tri à bulles**

* **Principe :**  
  Compare successivement les éléments adjacents et les échange si nécessaire. À chaque passage, le plus grand élément se déplace ("bulle") vers la fin du tableau. Le processus se répète jusqu’à l’absence d’échange.

**III. Tri rapide (Quick Sort)**

* **Principe :**  
  Sélectionne un pivot et divise le tableau en deux sous-tableaux : ceux inférieurs au pivot et ceux supérieurs. Chaque sous-tableau est trié récursivement. Le pivot se retrouve alors à sa position finale.

**IV. Tri par sélection**

* **Principe :**  
  Repère l’élément minimum dans la partie non triée et l’échange avec l’élément de la position actuelle. La portion triée s’agrandit progressivement au fur et à mesure que les plus petits éléments se placent en début de tableau.

**V. Tri de Shell**

* **Principe :**  
  Améliore le tri par insertion en commençant par trier des éléments éloignés (avec un grand « gap ») puis en réduisant progressivement cet écart. Le tableau devient de plus en plus ordonné jusqu’au tri final par insertion.

**VI. Tri par tas (Heap Sort)**

* **Principe :**  
  Transforme le tableau en un tas binaire (structure de type « heap ») où le plus grand (ou le plus petit) élément est à la racine. Ensuite, l’élément racine est échangé avec le dernier élément, et le tas est réajusté. Ce processus est répété jusqu’à ce que le tableau soit trié.

**VII. Tri fusion (Merge Sort)**

* **Principe :**  
  Divise récursivement le tableau en deux parties jusqu’à obtenir des sous-tableaux de taille 1, puis fusionne ces sous-tableaux de manière ordonnée. Cette méthode « diviser pour régner » garantit une bonne performance sur de grands ensembles de données.

**VIII. Tri comptage (Counting Sort)**

* **Principe :**  
  Compte le nombre d’occurrences de chaque valeur dans une plage définie. Ces comptes servent ensuite à placer directement chaque élément dans sa position finale dans le tableau trié. Méthode non comparative, efficace pour des valeurs dans une plage limitée.

**IX. Tri par casiers (Bucket Sort)**

* **Principe :**  
  Répartit les éléments dans plusieurs "casiers" selon leur valeur. Chaque casier est trié individuellement (souvent avec un algorithme simple), puis les casiers sont concaténés pour former le tableau final trié. Idéal pour des données uniformément réparties.

**X. Tri radix (Radix Sort)**

* **Principe :**  
  Trie les nombres chiffre par chiffre, en commençant généralement par le chiffre le moins significatif. À chaque passe, un tri stable (comme le tri comptage) regroupe les éléments selon le chiffre examiné, ce qui permet de traiter efficacement des données à longueur fixe.