

# Rapport sur l'Implémentation d'Algorithmes de Tri en RISC-V

## Introduction

Ce rapport analyse un programme assembleur RISC-V qui implémente et compare quatre algorithmes de tri classiques : tri à bulles, tri par sélection, tri par insertion et tri rapide (quicksort). Le programme visualise l'exécution des algorithmes sur une matrice de LED et mesure leurs performances en comptant le nombre d'instructions exécutées.

## Structure du Programme

Le code est organisé en plusieurs sections :

- Section des données : tableaux, constantes et compteurs
- Fonction principale (main) : orchestration des tests
- Implémentations des algorithmes de tri
- Fonctions auxiliaires : affichage, délai, mélange et réinitialisation du tableau

## Caractéristiques Principales

### 1. Visualisation en Temps Réel

Chaque algorithme affiche son fonctionnement sur une matrice LED, permettant d'observer visuellement les échanges d'éléments et les comparaisons pendant le processus de tri.

### 2. Mesure de Performance

Le programme compte le nombre d'instructions exécutées par chaque algorithme, fournissant une métrique objective pour comparer leur efficacité.

### 3. Réinitialisation et Mélange

Avant chaque test, le tableau est réinitialisé à son état d'origine puis mélangé aléatoirement pour garantir que chaque algorithme commence avec les mêmes conditions.

## Algorithmes Implémentés

### Tri à Bulles - Bubble Sort

- Principe : Compare les éléments adjacents et les échange si nécessaire
- Optimisation : Arrêt anticipé si aucun échange n'est effectué dans une itération
- Complexité moyenne :  $O(n^2)$

### Tri par Sélection - Selection Sort

- Principe : Recherche l'élément minimum dans la partie non triée pour le placer à la fin de la partie triée
- Complexité :  $O(n^2)$  dans tous les cas

### Tri par Insertion - Insertion Sort

- Principe : Construit le tableau trié en insérant chaque élément à sa position correcte
- Complexité moyenne :  $O(n^2)$ , mais efficace pour les tableaux presque triés

### Tri Rapide - Quick Sort

- Principe : Divise le tableau autour d'un pivot, puis trie récursivement les deux sous-tableaux
- Implémentation : Utilise l'élément à l'extrémité droite comme pivot
- Complexité moyenne :  $O(n \log n)$

### Aspects Techniques

1. **Visualisation** : Utilise l'appel système (ecall) 0x100 pour afficher les éléments sur la matrice LED
2. **Délais** : Ajoute des pauses entre les opérations pour faciliter l'observation
3. **Comptage d'Instructions** : Incrémente un compteur à chaque itération de boucle principale
4. **Gestion de la Mémoire** : Sauvegarde et restaure correctement les registres lors des appels de fonction
5. **Algorithme de Mélange** : Utilise l'algorithme de Fisher-Yates avec un générateur congruentiel linéaire pour les nombres pseudo-aléatoires

### Conclusion

Ce programme offre une excellente démonstration des algorithmes de tri classiques. Il permet de comprendre visuellement leur fonctionnement et d'analyser quantitativement leurs performances. Les résultats confirment que le tri rapide est généralement plus efficace que les algorithmes de complexité  $O(n^2)$  pour les grands ensembles de données.