

# Laboratoire 2 - Bogosort

## Table des matières

- [Table des matières](#)
- [Introduction au problème](#)
- [Choix d'implémentation](#)
  - [Gestion des threads](#)
  - [Calcul du pourcentage effectué](#)
  - [Mise à jour de la barre de progression](#)
- [Tests effectués](#)
  - [Tri](#)
  - [Affichage](#)
  - [Reset](#)
- [Chapitre exemple](#)

## Introduction au problème

Le problème premier de ce laboratoire est d'ajouter la partie multi-thread au Bogosort. Ceci est nécessaire pour avoir des temps de calcul raisonnable car le Bogosort est très peu efficace.

## Choix d'implémentation

Nos choix d'implémentation se séparent en plusieurs catégories.

- Gestion des threads
- Calcul du pourcentage effectué
- Mise à jour de la barre de progression

## Gestion des threads

Pour la gestion des threads le format est assez basique, créer tous les threads demandés, laisser effectué le bogosort passé en paramètre et supprimer tous les threads.

Ce qui a requis plus de réflexion est la récupération du tableau trié ainsi que l'arrêt des autres threads n'ayant pas encore trouvé la solution.

Pour ça nous avons passé en paramètre du bogosort, à la création de chaque thread, une variable partagée pour stocker le résultat.

Lorsqu'un thread trouve la séquence triée, il "l'annonce au thread manager" en changeant la valeur d'une variable partagée et ceci arrête les autres threads qui obtiennent l'information que la solution a été trouvée.

Le thread ayant trouvé la solution la met dans la variable partagée pour récupération par le thread manager.

## Calcul du pourcentage effectué

Le calcul du pourcentage se fait à chaque nouvelle permutation et consiste à contrôler combien des éléments sont placés correctement par rapport à leur voisin direct.

Ceci est effectué en itérant à travers la séquence et en augmentant un compteur à chaque fois qu'une valeur est plus grande que le précédent.

Ceci veut dire que la progression totale dépend de l'addition de plusieurs petites parties de la séquence au contraire du placement finale des éléments qui serait beaucoup plus long à calculé et reviendrait à trier la séquence.

## Mise à jour de la barre de progression

La barre de progression est mise à jour à chaque fois qu'un thread trouve une séquence plus trié que la précédente plus triée.

Ceci se fait en comparant le pourcentage trié par rapport à la valeur partagée dernièrement modifiée.

Une fois qu'une séquence plus triée est trouvée, le pourcentage est envoyé au thread manager qui lui émet un signal au mainWindow pour mettre à jour l'affichage.

Le signal envoyé doit être la différence par rapport au dernier affichage donc l'incrément est calculé en la différence entre la dernière plus triée et la valeur actuelle.

## Tests effectués

### Tri

Il a fallu contrôler que l'algorithme de tri implémenté fonctionne réellement.

Pour ceci nous avons effectué de nombreux tri en contrôlant le résultat final.

Nous avons aussi contrôler le calcul du pourcentage effectué par un procédé similaire.

En mode debug nous nous sommes arrêté après le calcul du pourcentage et avons contrôlé la valeur par rapport à la séquence.

### Affichage

Lors du tri nous avons vérifié que la barre de progression s'incrémente petit à petit et que les valeurs correspondent réellement à la séquence la plus triée.

### Reset

Lorsqu'un tri est fini et que la solution a été donnée il faut pouvoir démarrer un autre tri sans fermer l'application.

Pour ceci nous avons contrôlé que les variables contrôlant le pourcentage calculé sont bien remises à zero pour que l'affichage soit correct pour chaque tri.