

# Déroulement et évaluation

14h TPs

## 1) Choix du problème

Dans un premier vous devez choisir l'un des deux problèmes à résoudre : Optimisation de tournées touristiques ou Conception d'emploi du temps.

## 2) Conception d'une ou de plusieurs méthodes de résolution

Vous devez ensuite concevoir « sur papier » une méthode de résolution en vous appuyant sur le cours. Vous devez donc réfléchir à l'encodage d'une solution, le type de la méthode, les opérateurs de la méthode, etc. Pensez à mettre proprement sur papier toutes vos idées (ça vous aidera pour la partie évaluation).

## 3) Développement

Puis, vous allez développer en C++ vos algorithmes afin de résoudre efficacement le problème. Vous devez reprendre les classes Solution et Instance ainsi que le fichier main.cpp. Les classes Solution et Instance ne doivent absolument pas être modifiées (si vous souhaitez une modification, vous devez me la faire valider). Une vidéo est disponible pour expliquer ces deux classes et le main.cpp. Il n'est pas nécessaire de comprendre Instance.cpp et Solution.cpp. Si vous souhaitez reprendre le codage de la classe solution, vous devez soit recopier les attributs liés au codage soit l'imbriquer dans votre propre classe de codage de solution. La classe instance sert à manipuler les données. La classe solution sert à vérifier la validité et faisabilité d'une solution (très utile pour le débogage). Le fichier main.cpp doit être le main de votre programme. La fonction main ne doit pas être modifiée sans accord de ma part. Vos méthodes de résolution doivent être appelées dans la fonction « Resolution » qui doit absolument retourner la meilleure solution que vous avez trouvée. Avant de retourner cette valeur vous devez également absolument tester cette meilleure solution en utilisant la classe solution (et sa fonction vérification) que je vous ai fournis. Les trois #define du fichier main.cpp servent à indiquer le dossier qui contient les données, le nom du fichier qui indique l'ensemble des instances à résoudre (la fonction Resolution sera appelée pour chaque instance) et le fichier de sortie. Ce fichier de sortie permet de connaître pour chaque instance le temps de résolution et la fonction objectif de la meilleure solution connue.

## 4) Tester et analyser

Testez vos algorithmes sur les données disponibles et essayez d'améliorer vos algorithmes pour obtenir le meilleur compromis entre temps de résolution et qualité de la solution obtenue. Un fichier Resultats.xlsx vous permet d'avoir une idée sur la valeur des meilleures solutions.

## 5) Concours

Afin de gagner le super lot en étant premier au classement final sur l'un des deux problèmes, vous devez obtenir les meilleurs résultats sur les instances disponibles et sur des instances non disponibles (un tiers environ). Le classement sera basé sur le nombre de fois que votre méthode trouve la meilleure solution sur une instance en moins de 3 minutes.

## 6) Livrable et Évaluation

A la fin du projet, vous devez rendre l'ensemble de votre code source (qui doit être opérationnelle : compilable et exécutable) par mail [yannick.kergosien@univ-tours.fr](mailto:yannick.kergosien@univ-tours.fr) **avant le 26 avril 23h59min et 59s** (pénalité de -2 pts par jour de retard). Des entretiens seront ensuite organisés. Lors de cet entretien, vous devez présenter vos méthodes de résolution et votre code (préparez vos notes personnelles).

Le barème sera le suivant :

- Propreté du code (2 pts)
- Pertinence/originalité de la méthode (8 pts)
- Complexité de la méthode (5 pts)
- Efficacité de la méthode (2 pts)
- Clarté de la présentation lors de l'entretien (2 pts)
- Auto-critique de la méthode (1 pt)

Enfin des pénalités seront données si votre code ne compile pas (-2pts par bug corrigé).