

Projet ING2 : Optimisation d'une station de ski

Plan du projet

1. Objectifs et planning du projet	2
2. Introduction	3
3. Consignes du projet	4
4. Cahier des charges de M. Touchousse	5
4.1 Synthèse du plan du site	5
4.2 Classes à modéliser	6
4.3 Algorithmes de base	6
4.4 Analyse combinatoire et algorithmique du plus court chemin	6
4.5 Optimisation des vacances	6
4.6 Extensions : étude des flots de skieurs et autres extensions	7
5. Contraintes matérielles et logicielles	7
6. Critères d'évaluation	8
Barème	8
7. Modalités de dépôt sur campus	8

1. Objectifs et planning du projet

Ce second projet informatique a cette fois pour but de travailler la **Théorie des graphes** développée dans le **langage orienté objet C++, intégrée à un environnement console** (pour la saisie et l'affichage) et optionnellement graphique.

Le projet en mode "piscine" de Théorie des graphes se déroule **dans la semaine du 5 avril 2021** (1 semaine complète). Le Cahier Des Charges (CDC) du sujet vous est fourni la veille du démarrage des longueurs de nage. Deux conséquences pour vous :

- Il vous sera nécessaire d'étudier attentivement et rapidement le CDC pour le décomposer en étapes, identifier celles que vous pouvez traiter dès maintenant et commencer à les réaliser (comme vue en cours avec la méthode Pert).
- Pour les autres tâches, vous aurez également à identifier les besoins et à rechercher les connaissances nécessaires pour y répondre. Pour cela, vous aurez à utiliser tous les outils à votre disposition (web, tutoriaux, livres...) pour identifier, comprendre et vous approprier les techniques utiles au jeu et qui ne vous auraient pas été encore enseignées. Ainsi, à l'occasion des futurs cours, vous aurez les idées claires, un besoin précis, de nombreuses questions, et surtout une capacité accrue de comprendre la théorie enseignée.

Ce projet est à faire en **binôme ou trinôme issu du même groupe de TD, telle que mentionnée dans le lien** [Drive des équipes du projet Théorie des graphes](#)

Un « **suivi de projet** » pourra être organisé durant cette semaine « piscine » afin de répondre à vos questions.

Les dépôts du code et de la documentation PowerPoint rapport sur campus sont à effectuer dans les liens suivants de la page campus du « Projet d'info4 Théorie des graphes » et la dernière section **Livrables à déposer** : voir les liens [Livrable à déposer : code en C++ deadline le 11 avril](#) et [Documentation à déposer : PowerPoint \(ou Prezi\) deadline le 11 avril](#)

Les **soutenances** auront lieu **la semaine du 12 avril 2021**. Elles seront de **15 minutes au maximum par équipe et 5 minutes de questions posées par le jury**. Chaque équipe présentera son projet déposé sur campus par visio sur teams devant les autres équipes de son groupe de TD. Chaque étudiant d'une équipe doit intervenir lors de la présentation orale et avoir une connaissance globale du projet de l'équipe. Le jury pourra interroger l'équipe et individuellement certains étudiants. Un étudiant trop hésitant sur une question le concernant ou n'ayant pas une vision globale du projet pourra se voir pénaliser par rapport à ses coéquipiers. Tous les étudiants du groupe **doivent être présents sous peine d'être pénalisés par 0 au projet, toujours selon Pert** 😊

2. Introduction

En prévision du prochain SDI qui se tiendra dans une grande station de ski de Savoie (les Arcs pour ne pas la nommer), son directeur M. Touchousse, en accord avec l'ECE, cherche à optimiser la gestion de son entreprise tout en rendant le domaine skiable plus attractif auprès des touristes.

Passionné d'informatique, il décide d'engager les étudiants chevronnés d'ING2 (dont vous faites partie !). M. Touchousse fait part de son idée d'installer des bornes interactives dans les files d'attente des remontées et dans divers points de la station permettant aux skieurs de sélectionner son itinéraire. Il souhaitait notamment leur indiquer comment se rendre le plus rapidement possible vers n'importe quel autre point de la station.

M. Touchousse s'adresse alors aux professeurs de l'ECE et demande ironiquement : « - Mais vos petits génies en informatique seraient-ils capables de m'écrire un logiciel calculant rapidement un tel itinéraire ? ».

Les profs répondent d'une seule voix : « Bien sûr ! Ils vont modéliser votre station comme un graphe orienté où les sommets représentent les points de la station et les arcs représentent les pistes et les remontées. Les temps de descente et de montée étant évidemment non négatifs c'est ensuite un jeu d'enfant d'appliquer au graphe l'algorithme de Dijkstra qui vous donnera votre résultat en $O(n^2)$! »

Le plan graphique des pistes est téléchargeable dans le fichier [les arcs.bmp](#) en section **Projet "piscine" : planning et sujet** de la page campus du « Projet d'info4 Théorie des graphes ».



3. Consignes du projet

Respectez scrupuleusement le [cahier des charges de M. Touchousse](#), sous peine qu'il vous mette hors piste.

Respectez les consignes spécifiées dans la dernière section [Livrables à déposer](#) de la page campus du « Projet d'info4 Théorie des graphes ».

- Servez-vous du fichier de synthèse [data_arcs.txt](#) dont les informations sont détaillées au [chapitre 4.1](#) du présent document. Vous **devez mettre à jour ce fichier** selon les nouvelles données qui vous sont fournies dans ce sujet : voir le [cahier des charges de Mr Touchousse](#).
- Dans un premier temps, modélisez les [classes à modéliser au 4.2](#).
- Puis implémentez les [algorithmes de base du chapitre 4.3](#).
- Améliorez encore le projet en faisant [l'analyse algorithmique et combinatoire du plus court chemin au chapitre 4.4](#).
- Ensuite comme le souhaite M. Touchousse, [optimisez vos vacances au chapitre 4.5](#).
- Pour les plus ambitieux, des [extensions étudiant les flots de skieurs au chapitre 4.6](#) sont possibles. Et bien d'autres extensions issues de votre imagination féconde.
- Les classes et les algorithmes de base sont **à coder en C++**.
- Séparer le chargement des données des méthodes/algos de traitements et des saisies/affichages
- Pour les plus doués d'entre vous en mode graphique Allegro, vous pouvez développer une interface graphique événementielle, performante et intelligente pour la borne interactive avec Allegro ou autre librairie graphique (SDL, SFML...) compatible C++, en utilisant le plan de la station fourni dans le fichier [les_arcs.bmp](#). **L'interface graphique avec Allegro ou toute autre librairie graphique ne constitue pas du tout une priorité dans ce projet purement Théorie des graphes en C++, et ne sera pas considéré comme un bonus. Toutes les saisies et les affichages sont à faire en priorité en mode console.**
- Consignes sur la documentation PowerPoint (ou Prezi) à rédiger (environ 10 slides) :
 - 1) Page de garde avec titre, groupe de TD, numéro d'équipe et noms (1 slide)
 - 2) Sommaire (1 slide)
 - 3) **Diagramme de classes** (1 slide) avec l'outil [Draw.io](#) ou équivalent, et présentant les attributs, les méthodes, les cardinalités, MAIS sans constructeurs ni getters/setters
 - 4) **Versioning GIT en bonus** : screenshot montrant la bonne utilisation et répartition des tâches du code entre coéquipiers. (1 slide)
 - 5) **Méthodologie de votre travail** : (environ 5 slides)
 - tableau de la répartition des tâches et **l'ordonnancement du projet avec la méthode Pert**
 - présentation synthétique de vos approches (inspirez-vous des slides du cours avec des algos courts et "en français")
 - tableau comparatif de la complexité des différents algorithmes
 - La mise à jour commentée et argumentée du fichier de synthèse [data_arcs.txt](#) avec des exemples concrets
 - remarques, schémas, diagrammes, formules mathématiques justifiées, screenshots annotés : tout ce qui peut permettre de rendre votre présentation informative, efficace et dynamique
 - présentation claire, concise et précise du travail réalisé
 - > qu'avez-vous entrepris ? -> comment ? -> quels sont les résultats ?
 - 6) **Bilan individuel et collectif sans blabla** de l'état du travail effectué. (1 slide)
 - 7) **Bibliographie précise de toutes vos sources (web, livres etc.).** (1 slide). **Toute source non citée est considérée de facto comme un plagiat**

4. Cahier des charges de M. Touchousse

4.1 Synthèse du plan du site

Le plan du site est synthétisé dans deux fichiers que vous téléchargerez de la page campus <https://pedago-ece.campusonline.me/course/view.php?id=5493> du « Projet d'info4 Théorie des graphes » :

- Une image bitmap [les_arcs.bmp](#) contenant le plan graphique du site
- Un fichier texte [data_arcs.txt](#) contenant les informations suivantes :
 - <Nombre de points> puis, pour chaque point :
 - <n° point> <nom du lieu> <altitude>
 - <Nombre de trajets> puis, pour chaque trajet :
 - <n° trajet> <nom du trajet> <type> <n°point départ> <n°point arrivée>

Pour les trajets on distingue les types suivants : **TK** pour un télési, **TS** pour un télésiège, **TSD** pour un télésiège débrayable, **TC** pour une télécabine, **TPH** pour un téléphérique, **BUS** pour une navette, **V** pour piste verte, **B** pour piste bleue, **R** pour piste rouge, **N** pour piste noire, **KL** pour la piste de Kilomètre Lancé et **SURF** pour le domaine réservé au surf.

L'étude expérimentale effectuée pendant votre séjour dans la station vous a permis de définir les règles suivantes pour le calcul des temps de trajet :

- Pour les descentes, le temps est proportionnel au dénivelé.
 - Piste verte (**V**) 5mn pour 100m de dénivelé.
 - Piste bleue (**B**) 4mn pour 100m de dénivelé.
 - Piste rouge (**R**) 3mn pour 100m de dénivelé.
 - Piste noire (**N**) 2mn pour 100m de dénivelé.
 - Piste de kilomètre lancé (**KL**) 10sec pour 100m de dénivelé.
 - Snowpark (**SURF**) 10mn pour 100m de dénivelé.
- Pour les remontées, le temps est composé d'une partie fixe et d'une partie proportionnelle au dénivelé.
 - Téléphérique (**TPH**) 4mn + 2mn pour 100m de dénivelé.
 - Télécabine (**TC**) 2mn + 3mn pour 100m de dénivelé.
 - Télésiège débrayable (**TSD**) 1mn + 3mn pour 100m de dénivelé.
 - Télésiège (**TS**) 1mn + 4mn pour 100m de dénivelé.
 - Télési (**TK**) 1mn + 4mn pour 100m de dénivelé.
- Pour les navettes (**BUS**), en comptant le temps d'attente, le trajet dure en moyenne 40mn pour aller d'Arc2000 à Arc1600 ou d'Arc1600 à Arc2000 et 30mn pour aller d'Arc1600 à Arc1800 ou d'Arc1800 à Arc1600

- ❖ Les informations présentes dans le fichier servent de base de données à votre plan du site. Tout en les respectant, vous devrez les enrichir à chaque nouvelle règle : par exemple, les temps de trajet par défaut fournis ci-dessus sont à ajouter dans le fichier.
- ❖ Ces temps par défaut pourront être modifiés de manière interactive par l'utilisateur grâce à une interface console événementielle.

Toute autre information que vous jugez nécessaire peut être ajoutée dans ce fichier. Par exemple, vous pouvez ajouter si vous le souhaitez les coordonnées de chaque point de votre plan graphique.

4.2 Classes à modéliser

❖ **Votre première mission, car vous l'acceptez, est de modéliser le diagramme des différentes classes pour représenter :**

- un point de la station,
- un trajet (piste, remontée ou tronçon de route entre deux arrêts de bus, à ne pas confondre avec un chemin) en prenant en compte les contraintes sur leur durée, indiquées ci-dessus.
- la station, avec le graphe des trajets
- etc.

4.3 Algorithmes de base

❖ **Les premiers algorithmes à justifier dans votre documentation et à coder en C++ sont les suivants :**

- Implémentez le chargement du fichier texte fourni par M Touchousse : vous mémoriserez d'abord l'ensemble des points.
- Justifiez vos choix algorithmiques, en particulier les listes d'adjacence ou les listes de voisins dans l'implémentation du graphe des trajets.
- Implémentez une interface simple d'aide en **mode console** à l'utilisateur permettant :
 - en choisissant un trajet (piste ou remontée ou entre 2 arrêts de bus), de savoir d'où il part et où il arrive,
 - en choisissant un point, de savoir les trajets qui en partent et qui permettent d'y arriver.

4.4 Analyse combinatoire et algorithmique du plus court chemin

❖ **Votre mission, car vous l'acceptez, est maintenant d'analyser dans votre rapport les aspects combinatoires et algorithmiques permettant d'implémenter en C++ :**

- un algorithme déterminant, à partir d'un point de départ, tous les plus courts chemins issus de ce point ainsi que leurs temps.
- un algorithme qui affiche **en mode console** l'itinéraire le plus rapide entre deux points sur la borne interactive.

Ajoutez à votre interface ce qu'il faut pour choisir deux points de la station et afficher l'itinéraire le plus rapide obtenu par la méthode précédente.

4.5 Optimisation des vacances

Ayant réalisé avec brio la mission que vous avait confiée M. Touchousse, celui-ci vous offre une semaine de vacances aux Arcs. Vous avez remarqué que le programme que vous lui avez fourni n'est pas très intéressant pour vous : vous êtes en vacances et vous avez donc tout votre temps pour aller d'un endroit à un autre de la station. De plus, vous avez remarqué que les chemins les plus rapides sont rarement les plus intéressants.

Vous n'avez pas envie de passer vos vacances sur les remontées et vous décidez donc que celles-ci auront un intérêt négatif. En revanche, l'intérêt associé à chaque piste (descente) est accru. Pour cela vous devez paramétrer ces remontées et ces descentes de manière interactive.

Vous décidez donc, avant de partir en vacances, d'ajouter rapidement une fonctionnalité à votre programme qui vous permettra de calculer le chemin le plus intéressant entre deux points de la station. Par exemple, demander les plus courts chemins n'empruntant que certaines catégories de pistes et/ou de remontées (un skieur moyen peut ne pas vouloir s'embarquer sur des pistes noires, certains ont peur des télésièges etc).

- ❖ **Ajoutez ces nouveaux paramètres à votre borne interactive en justifiant leur choix dans le rapport. Les valeurs de ces paramètres seront sauvegardées dans le fichier [data_arcs.txt](#) téléchargé initialement, ceci afin d'être ultérieurement chargées.**
- ❖ **Modifiez les classes et les algorithmes des étapes précédentes afin d'y intégrer ces nouveaux paramètres. Vous justifierez vos choix dans le rapport.**

4.6 Extensions : étude des flots de skieurs et autres extensions

Bravo. Vous avez encore rempli brillamment la mission que vous avait confiée M. Touchousse qui vous offre le snow-board dernier cri.

Les petits logiciels interactifs à l'usage des skieurs que vous avez réalisés fonctionnent mais son projet est maintenant autrement plus ambitieux : il souhaite améliorer le séjour des skieurs, de leur faire plaisir, qu'ils profitent au max de leurs séjours, qu'ils aient envie de revenir.

Pour cela, on peut étudier les flots pour connaître les remontées dont il faudrait augmenter le débit (capacité) du nombre de skieurs par heure, ou ajouter pour l'accès à certaines pistes (descentes).

A titre d'exemple, les débits (capacités) des remontées pourraient être les suivants:

- Téléphérique (TPH) : 1200 skieurs/heure
- Télécabine (TC) : 2200 skieurs/heure
- Télésiège débrayable (TSD) : 2500 skieurs/heure
- Télésiège (TS) : 1800 skieurs/heure
- Télési (TK) : 800 skieurs/heure
- Navettes (BUS) : 300 skieurs par heure

N'ayant pas de mission précise pour l'instant mais vous souvenant de vos cours de graphe, vous décidez astucieusement d'anticiper celle-ci en préparant le terrain pour la résolution des problèmes du flot maximum :

« Le but est d'optimiser le flot de skieurs circulant entre deux points, en remontée ou en descente. Puis connaissant un flot réalisable entre deux points, il sera sans doute intéressant de savoir trouver un plus court chemin dans le graphe d'écart correspondant ou les plus longs circuits en termes de glisses ou encore la plus longue descente (monter le plus haut et redescendre le plus bas) » vous dites-vous.

- ❖ **Ajoutez ces débits à votre borne interactive et sauvegardez-les dans le fichier. Vous indiquerez dans votre rapport le débit (capacité) que vous proposez pour les descentes.**
- ❖ **Ajouter le flot dans vos classes.**
- ❖ **Ecrivez l'algorithme de résolution du flot maximum : cet algorithme permet de calculer l'augmentation possible du flot entre deux points.**
- ❖ **Affichez le graphe d'écart des chemins trouvés en testant la valeur du flot de chaque arc du graphe réel par rapport à sa capacité. Puis calculez et affichez le plus court chemin en nombre d'arcs entre deux points dans le graphe d'écart.**
- ❖ **Toute autre extension est la bienvenue : par exemple, exclusion des pistes (selon leur couleur ou longueur), le moyen de glisser le plus longtemps en prenant le moins de remontée possible, arriver à la pause-déjeuner sans prendre de retard etc. Je fais confiance à votre créativité et à votre esprit de futurs ingénieurs pour enrichir votre borne interactive.**

5. Contraintes matérielles et logicielles

Langage C++ en mode **console** en priorité, et éventuellement en utilisant la librairie graphique Allegro (ou SDL, SFML...) sur **Code Blocks**.

Méthodes de conduite de projet : Orientée objet

Matériel :

Vos PC personnels ou ceux de l'école sur une plateforme **Windows**. La soutenance se fera à distance sur teams après récupération du ZIP déposé sur CAMPUS (**testez-le avant !**).

6. Critères d'évaluation

Votre travail sera jugé sur les critères suivants :

- Le respect rigoureux de toutes les consignes énoncées précédemment dans le [cahier des charges de Mr Touchousse](#)
- La bonne répartition des tâches entre les membres de l'équipe.
- L'intérêt, l'originalité, la clarté et toutes les caractéristiques que vous prendrez soin de mettre en avant lors de la soutenance.

Barème

- **Démo sur 20 points coefficient 2**
 - Premières missions des chapitres 4.1 à 4.5 **sur 16 points**
 - Extensions du chapitre 4.6 sur 7 points dont 4 points pour les flots comptabilisés **seulement si les premières missions sont accomplies**
- **Documentation PowerPoint (ou Prezi) sur 20 points coefficient 1**

7. Modalités de dépôt sur campus

Deux fichiers à déposer **en respect des consignes spécifiées sur campus** dans la dernière section [Livrables à déposer](#) de la page campus du « Projet d'info4 Théorie des graphes » :

- Dans le lien [Livrable à déposer : code en C++ deadline le 11 avril](#) : un fichier compressé du répertoire créé par l'équipe de projet aux normes « nom1-nom2-nom3.zip » avec toutes les sources, l'exécutable et un .exe d'installation autoextractible.
- Dans le lien [Documentation à déposer : PowerPoint \(ou Prezi\) deadline le 11 avril](#) : la documentation PowerPoint (ou Prezi) en **10 slides maximum** pour l'équipe de projet aux normes « nom1-nom2-nom3.doc (ou pdf) ».

Ne déposez pas à la dernière minute !!! Tout retard sera pénalisé de -2 points par jour de retard à partir du lendemain. Tout plagiat sera sévèrement pénalisé par 0 et un avertissement.

Bonne glisse à tous ☺

JPS et toute l'équipe pédagogique, et des remerciements particuliers à Mme Palasi pour ses conseils constructifs à votre attention, chers nageurs glisseurs ☺