

Contexte : préparation au concours ICFC

L'objectif de ce semestre est d'améliorer vos compétences et vos reflexes en algorithmique de manière ludique, en vous préparant au concours ICFC (non, on ne vous invite pas à la finale à Moscou 😊)

Le concours International Collegiate Programming Contest (ICPC : <https://icpc.baylor.edu/worldfinals/problems>) est un concours de niveau universitaire sur la programmation informatique. Lors de chaque édition, divers problèmes sont proposés. Par exemple, l'édition 2019 en contenait 11. A certains problèmes sont associés des temps limites, par exemple 10 secondes. Il s'agit du temps maximum que doit mettre votre programme pour résoudre le problème demandé. A certains problèmes sont également associés des input et des output. Les inputs correspondent aux informations qui seront données à votre programme et les outputs aux sorties, vous veillerez donc à construire les parsers adaptés. Un exemple de problème est donné en annexe.

Consignes

La préparation s'effectue en trinôme (tirage aléatoire des groupes). Vous vous répartirez les problèmes à préparer et suivrez l'avancement à l'aide de la méthode agile SCRUM.

Vous utiliserez Github avec des comptes nominatifs, pour vous partager le code. Pro-tip : ne jetez pas votre github après le projet ! Votre Github est un peu votre vitrine, avoir un joli Github sur votre CV avec vos projets est un plus pour vos futures candidatures.

Langage : C++ ou Python

Evaluation

Note de groupe (50%)

Vous rendrez les documents suivants :

- Un bilan de groupe : comment vous vous êtes organisés, les difficultés rencontrées, comment vous les avez surmontées, ce que vous avez appris,...
 - o Un tableau récapitulatif avec la liste des problèmes, et pour chacun :
 - Avez-vous réussi à le préparer ?
 - Quel est le temps d'exécution du programme pour ce problème ?
- Un bilan individuel : les tâches que vous avez réalisées, les difficultés rencontrées, comment vous les avez surmontées, ce que vous avez appris,...
- L'adresse du github où se trouvent les problèmes préparés, avec un accès pour M. Gouvy et Mme Jacques
- Vos rapports SCRUM (backlog et sprints backlogs)

Note individuelle (50%)

Deux séances d'évaluation sont prévues : une avec M. Gouvy et une avec Mme Jacques. Lors de ces séances, une année sera choisie aléatoirement parmi celles listées ci-dessous. Individuellement, vous traiterez le plus de problèmes possibles durant la séance. A la fin de la séance, vous rendrez votre code. Vous n'aurez pas accès aux documents durant l'évaluation.

Les annales sont accessibles sur : <https://icpc.baylor.edu/community/history>

14 avril avec Mme Jacques (4 séances préparatoires) : ICPC 1993 à 2003 (inclus)

19 mai avec M. Gouvy (6 séances préparatoires) : ICPC 2004 à 2019 (inclus)

Exemple de problème : détecteur de cul-de-sac (problème E 2019)

Temps maximum : 5 secondes.

Le maire de votre ville souhaite optimiser le placement des panneaux impasse dans sa ville, et en placer le nombre minimum requis. Le maire met à votre disposition un plan de la ville, sous la forme d'une liste de liaisons. Une liaison i et j signifie qu'il y a une rue entre les emplacements i et j . Deux exemples de fichiers d'input sont donnés ci-dessous, pour les deux exemples a et b. Votre objectif est de sortir un fichier output qui contient le nombre de panneaux à installer en 1^{ère} ligne, et la liste des liaisons où il faut installer un panneau. Lorsqu'il y a plusieurs impasses successives, comme dans l'exemple b), on disposera le panneau à la première rue concernée.

Dans l'exemple a), deux panneaux impasse sont nécessaires pour les rues 4->5 et 6->5. Dans l'exemple b) 3 panneaux sont nécessaires, pour les rues 1->5, 1->6 et 3->4. Pour économiser des panneaux, aucun panneau n'est proposé pour 6->8 et 6->7 car un panneau est déjà proposé pour 1->6

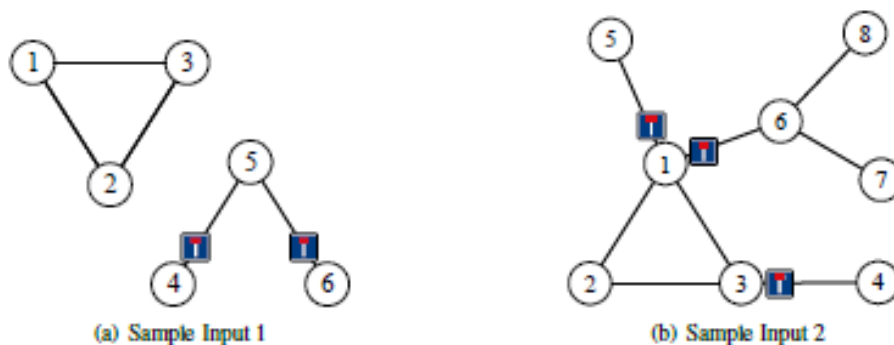


Figure E.1: Illustration of sample inputs, indicating where non-redundant dead-end signs are placed.

Sample Input 1	Sample Output 1
6 5 1 2 1 3 2 3 4 5 5 6	2 4 5 6 5

Sample Input 2	Sample Output 2
8 8 1 2 1 3 2 3 3 4 1 5 1 6 6 7 6 8	3 1 5 1 6 3 4