

Utilisation d'un juge en ligne pour l'évaluation automatique des programmes des étudiants

Thibault Raffailac, Francis Chavanon, Romain Vuillemot

LIRIS, École Centrale de Lyon, Université de Lyon, France

Abstract. L'enseignement supérieur de l'informatique met en oeuvre l'écriture de programmes complexes. Pour faciliter leur évaluation et favoriser l'autonomie des étudiants, nous proposons l'utilisation du juge en ligne DMOJ issu du domaine de la programmation compétitive. Nos objectifs sont de faciliter son utilisation pour l'enseignement en France, et d'étudier son influence sur l'implication des étudiants et leur aptitude à programmer. Nous présentons notre protocole de recherche ainsi que les premières observations de l'utilisation de DMOJ par des étudiants.

Keywords: apprendre à coder, autonomie, algorithmique, structures de données, programmation compétitive, outil numérique

1 Introduction

Dans le cadre des enseignements en algorithmique et structures de données, la préparation et la correction des Travaux Dirigés sont notoirement difficiles et chronophages. Un bon sujet de TD doit mettre en application les notions enseignées en cours, et produire des réponses qui puissent être évaluées facilement afin de s'assurer que chaque étudiant a correctement assimilé le cours.

La tâche est d'autant plus complexe lorsqu'il s'agit de produire du code informatique. Il est alors demandé aux étudiants d'écrire un ou plusieurs programmes pour répondre à une série d'exercices. Les enseignants sont là pour aider les étudiants à produire du code fonctionnel et correct, et éventuellement pour corriger les programmes à faire chez soi. Or chaque programme peut représenter plusieurs centaines de lignes de texte qu'il faut lire et exécuter systématiquement. Que ce soit durant un TD ou une correction a posteriori, ils nécessitent du temps et de la concentration, qui combinés au nombre d'étudiants font qu'il est difficile de consacrer suffisamment de temps à chacun. Les étudiants sont alors confrontés à un délai entre leurs requêtes (questions ou solutions) et les retours des enseignants, ce qui peut gêner leur rythme de travail.

2 Démarche

Dans le cadre de l'évolution des pratiques d'enseignement de l'algorithmique, et pour répondre à la problématique ci-dessus, nous proposons l'utilisation d'une plateforme d'évaluation en ligne (*online judge*) issue du domaine des compétitions

de code. Le domaine est aujourd’hui très populaire et attire à la fois les étudiants (pour les défis à relever), les universités (pour la visibilité dans les concours) et les entreprises (pour la sélection et le recrutement des meilleurs). Le format des compétitions de programmation respecte des conventions communes :

- Chaque compétition présente un ensemble de problèmes à résoudre en un temps imparti (de 2h à 24h).
- Chaque problème se résout en écrivant un programme, qui pour chaque occurrence de données d’entrée renvoie une solution en sortie.
- Les entrées/sorties de chaque programme sont spécifiées et utilisent les flux standard `stdin` et `stdout` d’Unix.
- Durant chaque compétition les participants soumettent leurs programmes à une plateforme, qui teste leurs résultats pour des données d’entrée cachées.
- Les entrées sont dimensionnées en fonction de la complexité des algorithmes attendus, interrompant par limite de temps les programmes trop lents.
- Une soumission est jugée valide si le programme compile correctement, s’exécute sans échec, n’est pas interrompu par limite de temps ou de mémoire, et donne la bonne réponse pour chaque test d’entrée.

Il existe de très nombreuses plateformes en ligne liées aux compétitions de programmation et ouvertes aux enseignants (voir [5] pour une classification récente complète). Or la majorité ne sont pas *open source* ou gratuites, voire ne sont plus maintenues, ce qui représente un risque si on investit du temps ou de l’argent et que l’outil ne répond finalement pas suffisamment à nos besoins. En outre, la plupart des plateformes existantes ne permettent pas de proposer ses propres problèmes. Il faut choisir dans une base d’exercices disponibles (conçus pour l’enseignement ou issus d’archives de compétitions), au risque de prendre des exercices non traduits ou trop difficiles. Nous avons donc sélectionné une plateforme *open source*, activement maintenue et très documentée, DMOJ [1] (voir Figure 1), que nous hébergeons localement.

3 Objectifs

Avec l’utilisation d’un juge en ligne, nous souhaitons d’abord automatiser et accélérer l’évaluation des différents TDs (lorsqu’ils se prêtent à l’expression en problèmes en ligne). Ensuite la plateforme permettra aux étudiants de s’entraîner hors des cours pour préparer les examens, avec des problèmes supplémentaires évalués automatiquement. Enfin nous souhaitons améliorer le suivi des étudiants, grâce à un accès complet aux programmes soumis et au profil en ligne de chacun.

À plus long terme le rapprochement au domaine de la programmation en compétition est un choix assumé [4]. Elle met en oeuvre plusieurs compétences en informatique, entre autres :

- Elle invite à raisonner en sélectionnant l’algorithme ou la méthode de programmation la plus appropriée (dans une longue liste), pour résoudre un problème le plus vite.

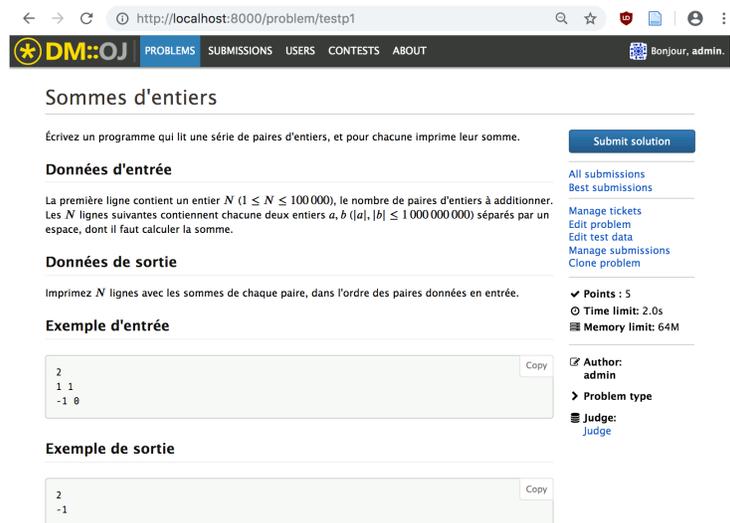


Fig. 1. Illustration de la plateforme DMOJ avec un problème simple.

- En proposant un grand nombre de problèmes à résoudre en un temps court, elle incite à ne pas surestimer leur difficulté (d’autant plus si d’autres les ont résolus plus rapidement que soi).
- Elle nécessite l’anticipation des cas difficiles (qui font toujours partie des données d’entrée cachées), car un juge en ligne ne communique jamais les données qui ont fait échouer un programme non valide.

4 Méthodologie

Dans un premier temps nous avons prévu d’intégrer l’utilisation de la plateforme de façon optionnelle par les étudiants (comme mesure de progression), afin de recueillir leurs retours pour améliorations. Les questions de recherche auxquelles nous souhaitons répondre sont les suivantes :

- Comment l’utilisation d’un juge en ligne va-t-elle influencer les tâches de préparation et évaluation des TDs ? (en temps consacré, et en qualité des retours des enseignants aux étudiants)
- Comment est-ce qu’il influencera le travail des étudiants en termes de compétence à l’examen, d’autonomie, de motivation et de satisfaction ?
- Comment est-ce qu’il influencera les disparités entre étudiants dans chaque groupe de TD ?

L’utilisation de juges en ligne pour l’enseignement est une démarche beaucoup étudiée dans la littérature [3, 2], or elle est également difficile à mettre en oeuvre. C’est pourquoi notre première contribution sera de faciliter la mise en place de DMOJ pour l’enseignement, en fournissant une image de machine

virtuelle prête à l’emploi et en documentant son utilisation. Pour répondre aux questions ci-dessus, nous nous baserons sur des entretiens avec les chargés de TDs, des questionnaires à destination des étudiants, leurs résultats aux différents examens, et les données de soumissions sur la plateforme DMOJ.

5 Premières observations et perspectives futures

Nous avons déjà mis en place la plateforme DMOJ avec le soutien du service informatique de notre université. Elle est aujourd’hui fonctionnelle et a été testée en conditions réelles par les étudiants. La première difficulté d’importance a été de l’intégrer avec les outils de travail des étudiants. En effet, nous enseignons la programmation avec Python sur l’IDE Spyder, or celui-ci ne permet pas de rediriger les entrées/sorties standard vers des fichiers, comme le feraient les chevrons `<` et `>` du terminal. L’utilisation du terminal n’étant pas au programme de nos enseignements d’informatique, l’utilisation de DMOJ s’intègre difficilement avec l’environnement de travail des étudiants, et a causé quelques confusions lors des premiers essais (y compris pour les enseignants).

Avec le contrôle sur l’apparence et les fonctionnalités de notre plateforme, nous pouvons intégrer des requêtes et évolutions vers d’autres usages. Il conviendra aussi de former les collègues enseignants à l’utilisation de DMOJ durant les TDs, et d’expliquer aux étudiants la nature des tests effectués sur la plateforme (données volumineuses, cas pathologiques). À l’avenir nous étudierons son utilisation pour les enseignements en recherche opérationnelle (qui se prêtent très bien à l’expression en problèmes en ligne), afin de motiver les étudiants avec des classements en temps réel. Au delà des compétences en programmation et résolution de problèmes, nous espérons à plus long terme améliorer le recrutement des étudiants dans les entreprises d’informatique, grâce à leur expérience des “challenges” courants des entretiens.

References

1. DMOJ: DMOJ, <https://github.com/DMOJ>
2. Enström, E., Kreitz, G., Niemelä, F., Söderman, P., Kann, V.: Five years with kattis — using an automated assessment system in teaching. In: 2011 Frontiers in Education Conference (FIE). pp. T3J-1–T3J-6. ISSN: 0190-5848, 0190-5848, 0190-5848, 2377-634X
3. García-Mateos, G., Fernández-Alemán, J.L.: A course on algorithms and data structures using on-line judging. In: Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education. pp. 45–49. ITiCSE ’09, ACM, <http://doi.acm.org/10.1145/1562877.1562897>, event-place: Paris, France
4. Vandeput, E., Henry, J.: Apprendre à programmer: comment les enseignants justifient-ils le choix d’un outil didactique? In: Didapro 7 – DidaSTIC. De 0 à 1 ou l’heure de l’informatique à l’école. <https://hal.archives-ouvertes.fr/hal-01753133>
5. Wasik, S., Antczak, M., Badura, J., Laskowski, A., Sternal, T.: A survey on online judge systems and their applications 51(1), 3:1–3:34, <http://doi.acm.org/10.1145/3143560>