

T.D/T.P 1 et 2 Tables Dynamiques

TD/TP1

- (1) Considérons que l'on ne double pas la taille de la table quand celle-ci est pleine, mais qu'on en multiplie la taille par un facteur $\alpha \geq 1$, donner une définition pour la fonction potentielle dans ce cas.
- (2) Donner le coût amorti de l'opération `Inserer-Table` en fonction de α .
- (3) Télécharger sur l'ENT le package, utilisez les outils développés pour afficher les coûts réels et amortis de l'opération `Inserer-Table`. Pour cela, suivez les instructions dans le fichier `README.md`. Compilez et exécutez les programmes dans les 4 langages, puis observez les résultats expérimentaux dans **gnuplot**.

Questions :

- (a) Lors de l'exécution des programmes, quelle est le morceau de code qui semble prendre le plus de temps à s'exécuter. Quelle est la complexité de ces fonctions ? Pourquoi ce morceau de code est il plus lent que le reste ?
 - (b) Observez maintenant le coût amorti en temps dans les différents langages. A quel moment le coût amorti augmente-t-il ? Pourquoi ? (La réponse dépend du langage).
 - (c) Affichez le nombre de copies effectués par chaque opération (C/C++ ou Java), puis le coût amorti. Que remarque t'on ? Quelle différence y a t'il avec le temps réel mesuré ?
 - (d) Recommencez plusieurs fois l'expérience avec les différents langages. Qu'est-ce qui change d'une expérience à une autre ? Qu'est ce qui ne change pas ?
 - (e) Tentez d'expliquer pourquoi certains langages sont plus rapides que d'autres dans cette expérience.
 - (f) Observez l'espace mémoire inutilisé au fur et à mesure du programme. Qu'en pensez vous ? Imaginez un scénario dans lequel cela pourrait poser problème.
- (4) Choisissez maintenant un langage de programmation en C, C++ et JAVA. Il est conseillé de choisir l'un de ces langages et de s'y tenir pendant tout le semestre. Attention, pour chacune des questions, il est conseillé de sauvegarder les affichages effectués sur **gnuplot** sous forme d'image afin d'en garder une trace. Les noms de fichiers d'images devront vous permettre de retrouver les conditions de l'expérience. Modifiez la fonction `do_we_need_to_enlarge_capacity` pour ne se déclencher que lorsque le tableau est plein. Relancer une expérience et commentez les résultats.
- (5) Dans la fonction `enlarge_capacity`, faites varier le facteur multiplicatif α . Que se passe-t-il ? Donner une règle, décrivant le rapport entre le coût en temps et le coût en espace.
 - (6) Dans la fonction `enlarge_capacity`, faites varier la capacité n vers une capacité $n + \sqrt{n}$. Que se passe-t-il ?