

Rapport de TP 4MMAOD : Justification optimale de texte

NOM Prénom étudiant₁ (groupe étudiant₁)

NOM Prénom étudiant₂ (groupe étudiant₂)

10 novembre 2018

Préambule Ce patron de rapport est donné à titre d'exemple pour répondre aux questions 4, 5 et 6 de l'énoncé. Vous pouvez soit compléter ce patron (complètement ou partiellement selon les résultats fournis par votre programme) soit rédiger votre propre rapport (qui ne doit pas excéder 2 à 3 pages). Dans tous les cas l'évaluation suit le barème indiqué ci-dessous.

- Compléter ce patron de rapport en supprimant toutes les phrases en italique qui ne doivent pas apparaître dans le rapport pdf.
- Le rapport doit tenir sur 2 à 3 pages au maximum.
- Barème (sur 20 points) : Programme=4 pts ; Tests=10 points ; Rapport= 6 points = 5 points + **1 point** pour la qualité globale du rapport : présentation, concision et clarté de l'argumentation.
- Détail du barème du programme sur 4 points : lisibilité du code=1 point ; commentaires/doxygen=1 point ; efficacité de la programmation=1 point ; gestion de la mémoire=1 point (mmap, allocation et libération)

1 Principe de notre programme (1 point)

Mettre ici une explication brève du principe de votre programme en précisant la méthode implantée (récursive, itérative) et les choix d'implémentation effectués (valeurs mémorisées, lecture du fichier (mmap?), écriture, etc).

2 Analyse du coût théorique (1.5 points)

Donner ici l'analyse du coût théorique de votre programme en fonction du nombre n de caractères, du nombre m de mots et de la longueur L d'une ligne. Pour chaque coût, donner la formule qui le caractérise (en justifiant brièvement pourquoi cette formule correspond à votre programme), puis l'ordre du coût en notation Θ de préférence, sinon O .

NB : Le travail (i.e. nombre d'opérations) est mesuré en nombre de comparaisons des coûts de justification (dans le calcul du Min de l'équation de Bellman).

2.1 Nombre d'opérations (comparaisons/min) en pire cas :

Justification :

2.2 Place mémoire requise :

Justification :

2.3 Nombre de défauts de cache sur le modèle CO :

Justification :

3 Compte rendu d'expérimentation (2.5 points)

3.1 Description de la machine et conditions expérimentales (0.5 point)

Décrire les conditions permettant la reproductibilité des mesures.

Décrire la machine utilisée pour les tests (par exemple `pcserveur.ensimag.fr` ou une autre machine auquel cas indiquer processeur et sa fréquence, la mémoire, le système d'exploitation).

Préciser si la machine était monopolisée pour un test ou si d'autres processus ou utilisateurs étaient en cours d'exécution.

Préciser comment les mesures de temps ont été effectuées (fonction appelée, par exemple `getrusage` ou la commande `unix time`) et l'unité de temps ; en particulier, préciser comment les 5 exécutions pour chaque test ont été faites (par exemple si le même test est fait 5 fois de suite, ou si les tests sont alternés entre les mesures, ou exécutés en concurrence etc).

3.2 Mesures expérimentales (1 point)

Compléter le tableau suivant par les temps d'exécution (temps minimum, maximum et moyen sur 5 exécutions) mesurés pour les longueurs de ligne ci-dessous sur le benchmark :

`/matieres/4MMA0D6/Benchmark/ALaRechercheDuTempsPerdu-1paragraphe-ISO-8859-1.in`

Seules les mesures pour lesquelles le programme marche doivent être indiquées. Si le benchmark ne passe pas, vous pouvez en prendre ou en construire un autre sur lequel votre programme fonctionne et avec des valeurs de M mettant en évidence des défauts de cache éventuels.

longueur ligne (M)	valeur de justification	temps elapsed min	temps elapsed max	temps elapsed moyen	temps user moyen	temps system moyen
200						
400						
600						
800						
1000						
1200						
1400						
1600						
1800						
2000						

FIGURE 1 – Mesures des temps minimum, maximum et moyen de 5 exécutions. Seules sont indiquées les valeurs de M significatives et pour lesquelles le programme s'exécute sans erreur et en temps raisonnable.

Compléter le tableau suivant par les mesures d'exécution données pour les longueurs M de ligne ci-dessous par la commande `valgrind` :

`valgrind -tool=cachegrind ./A0Djustify < M > ALaRechercheDuTempsPerdu-1paragraphe-ISO-8859-1`

longueur ligne (M)	#instructions (travail)	#défauts au total	#défauts en lecture	#défauts en écriture
200				
400				
600				
800				
1000				
1200				
1400				
1600				
1800				
2000				

FIGURE 2 – Mesures des défauts de cache avec `valgrind -tool=cachegrind`. Seules sont indiquées les valeurs de M significatives et pour lesquelles `valgrind` s'exécute sans erreur et en temps raisonnable.

3.3 Analyse des résultats expérimentaux (1 point)

Donner une réponse justifiée à la question : les mesures expérimentales correspondent-elles à l'analyse théorique (nombre d'opérations et défauts de cache) ?