

UE Ouverture : Devoir de Programmation

Edwin Ansari
Maria Sidko

1 Présentation

Question 1.3 On constate une complexité en $O(n)$ en nombre d'appels à `Random.int` et une complexité en $O(n^2)$ en nombre de match.

Question 1.6 On constate une complexité en $O(n \log n)$ en nombre d'appels à `Random.int` et une complexité en $O(n \log n)$ en nombre de match.

2 Compression des ABR

Question 2.11 Dans le pire des cas on constate une complexité en $O(n)$.

Question 2.12 On constate une complexité moyenne en $O(\log n)$.

3 Expérimentations : gains ou perte d'efficacité des ABR compressés

Question 3.13 Pour calculer le temps nous avons utilisé la différence entre deux appels à la fonction `Unix.gettimeofday()`.

Question 3.14 Nous avons fait une moyenne sur 5 recherches pour chaque n sur des arbres générés aléatoirement et avec un entier à rechercher généré aléatoirement, et nous obtenons le graphique de la Figure 1. On constate des temps très petits dans les deux cas, et à priori la compression ne ralentit pas beaucoup la recherche dans l'ABR. L'observation d'une réelle différence nécessiterait une expérimentation sur des arbres beaucoup plus grands ce que la puissance de la machine utilisée ne permet pas.

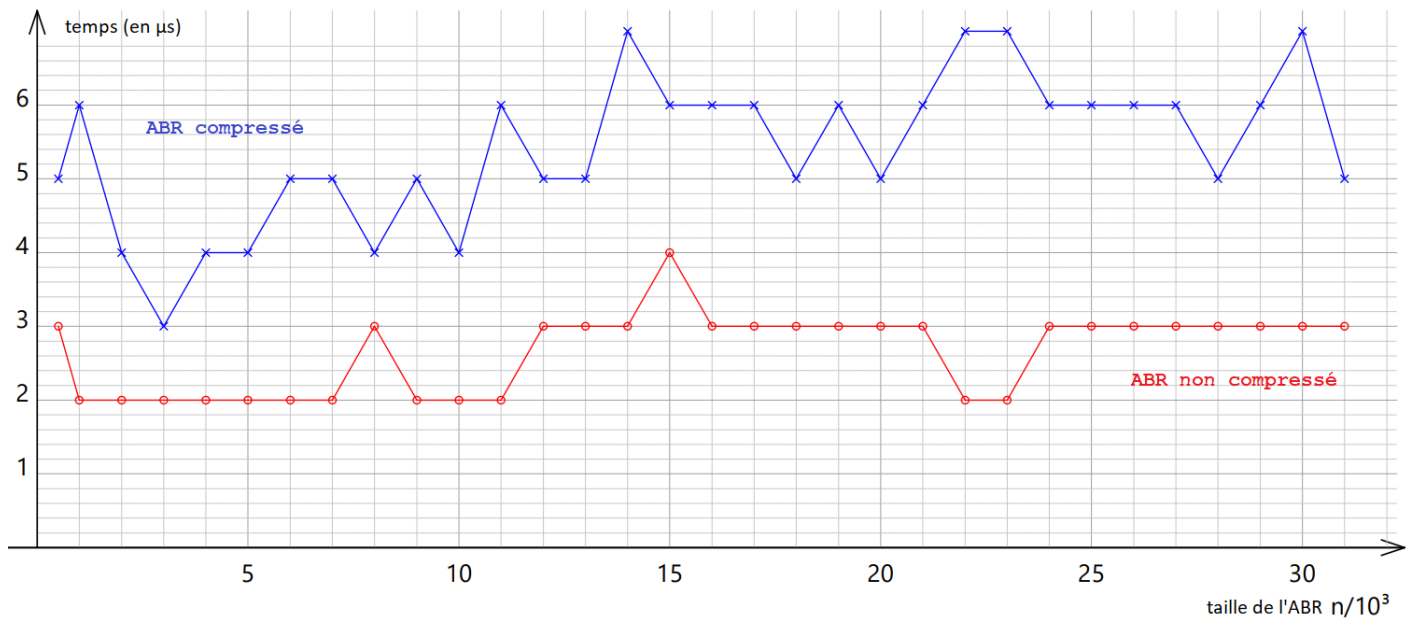


FIGURE 1 – Graphique des temps d'exécution de la fonction de recherche en fonction de la taille n de l'ABR

Question 3.15 En effectuant des appels successifs à la fonction pour des ABR compressés et non compressés à taille croissante générés aléatoirement nous obtenons le graphique de la Figure2. On constate en effet une diminution de l'usage de la mémoire suite à la compression de l'arbre.

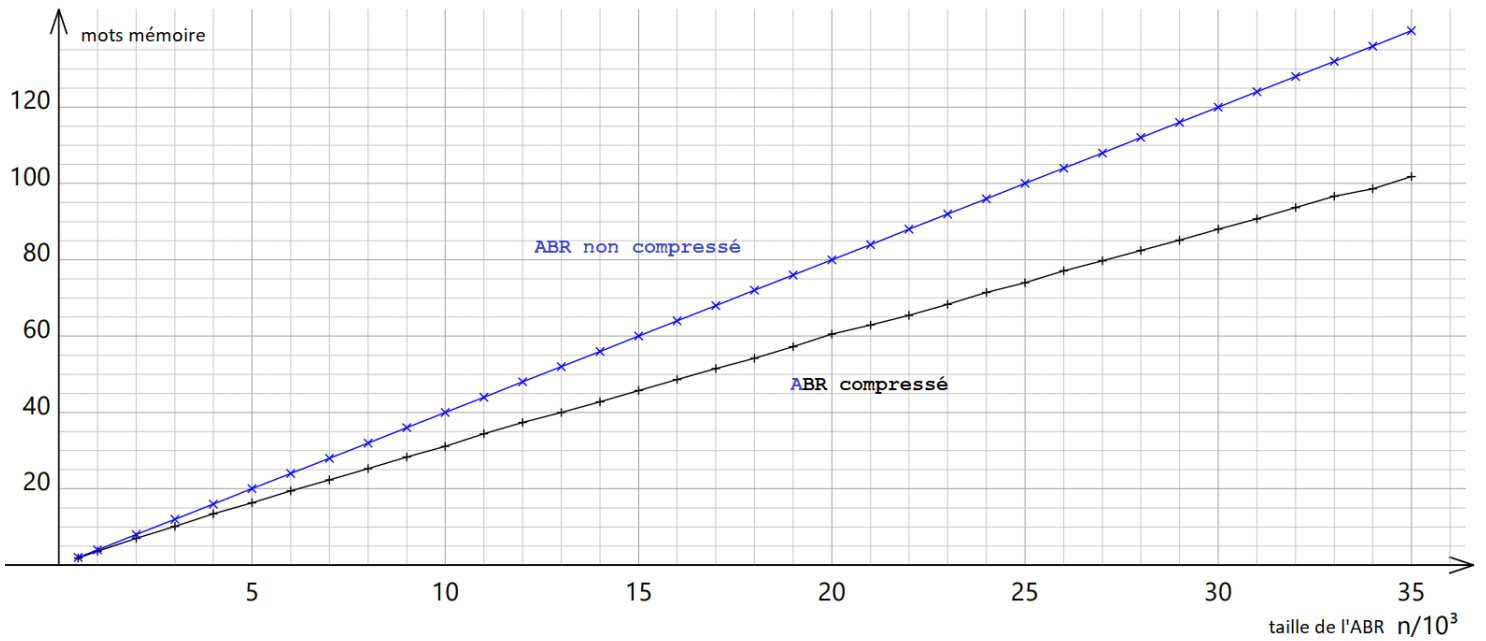


FIGURE 2 – Graphique du nombre de mots mémoire stockés sur le tas en fonction de la taille n de l'ABR