

Tapez-y la commande
gnuplot> plot ‘‘performance.txt’’
et vous obtiendrez un diagramme (plus d’info avec man gnuplot)

Exemple d’un programme utilisant la classe `std::chrono` pour faire des mesures de temps :

```
#include <chrono> //std::chrono::time_point
#include <cstdlib> /*atoi*/
#include <cstdio>

int main(int argc, char ** argv)
{
    int i,nbiter;
    if(argc!=2)
    {
        std::printf("Usage : \n %s suivi d'un entier \n
        (nombre de passages dans une boucle d'incrementation
        d'un int)\n", argv[0]);
        exit(1);
    }
    nbiter=std::atoi(argv[1]);

    std::chrono::time_point<std::chrono::system_clock> start, end;
    start = std::chrono::system_clock::now();
    for (i=1;i<=nbiter;++i) std::printf("%d %d\n",i, i*i);
    end = std::chrono::system_clock::now();
    int elapsed_microseconds
        = std::chrono::duration_cast<std::chrono::microseconds>(end-start).count();

    std::printf("Temps requis pour calculer %d carres : %d \n",
        nbiter, elapsed_microseconds);
    return 0;
}
```

2 Pousser plus loin les performances !

Si vous avez terminé d’enregistrer le profil des performances de vos `Skip-lists` (pour des valeurs de p variées), vous êtes armés pour les comparer à celles d’autres structures de données permettant de stocker une séquence triée.

2.1 Listes triées

Si vous ne l’avez pas encore terminé au TP2, mettez au point le module `ListeTriée` en modifiant le module liste chaînée du premier TP de manière à modifier son interface. L’idée était d’utiliser les listes chaînées pour y stocker des séquences triées. Du coup, les procédures d’ajout en queue et d’ajout en tête disparaissaient au profit d’une procédure d’insertion dans une liste triée. Enregistrez les performances de recherche et d’insertion d’un élément dans une `ListeTriée` de la même manière que vous l’avez fait avec les `Skip-lists`.

2.2 Comparaison des performances

Les profils que vous avez obtenus sont-ils cohérents avec les résultats vus en cours ? Vous pouvez s’il vous reste du temps établir des profils de temps d’insertion et de recherche dans un tableau dynamique trié (classe `std::vector` de la STL).