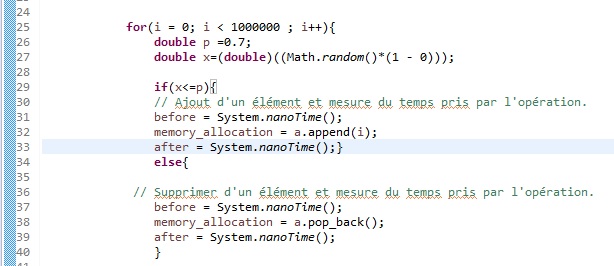
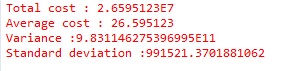
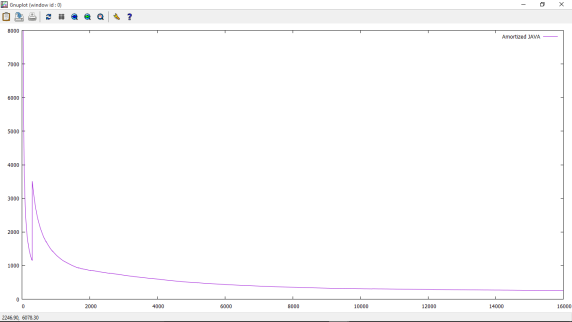
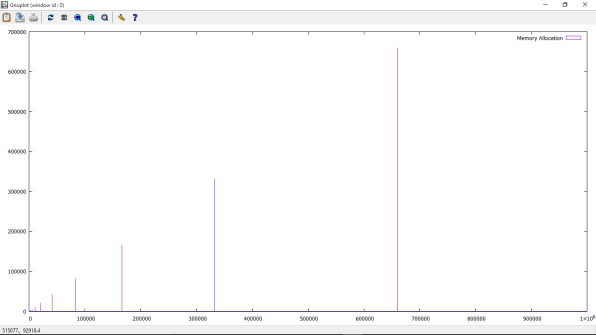
***TP2***

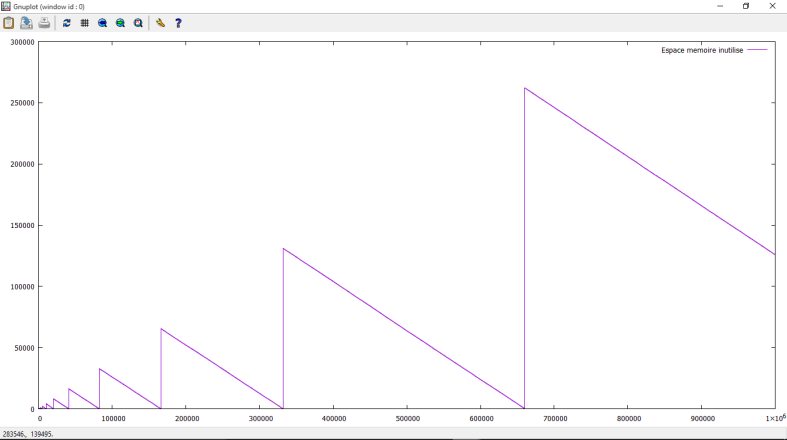
2.



3.

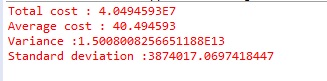


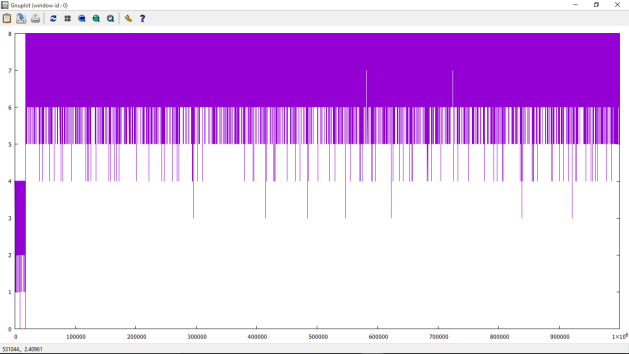
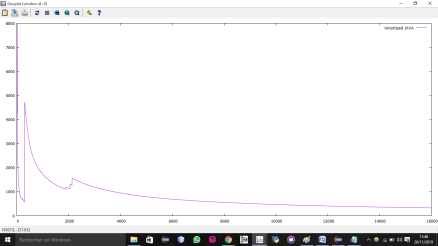
 

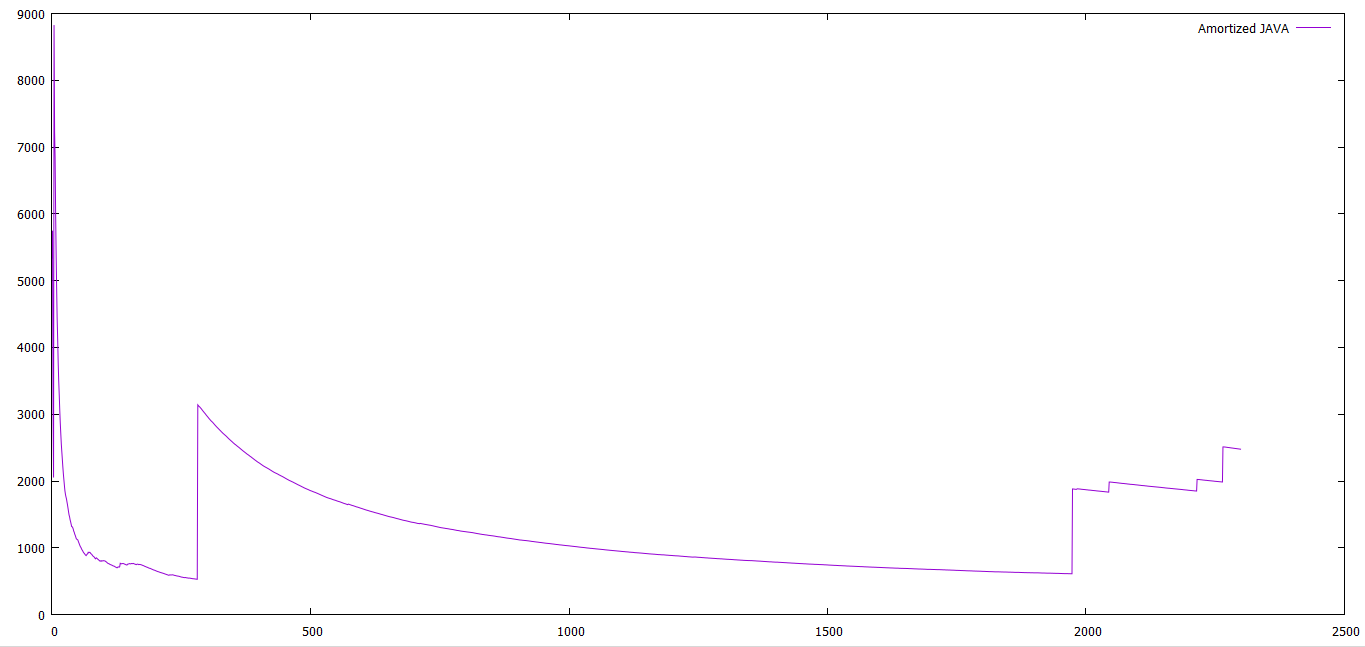


4.

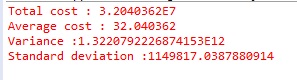
**0,1 :**

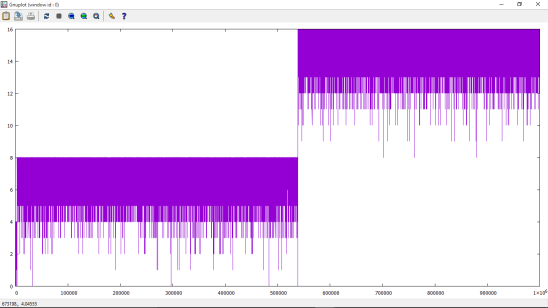
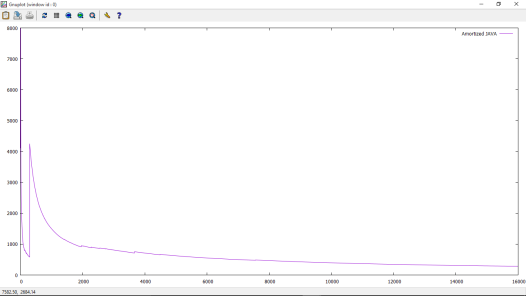
****

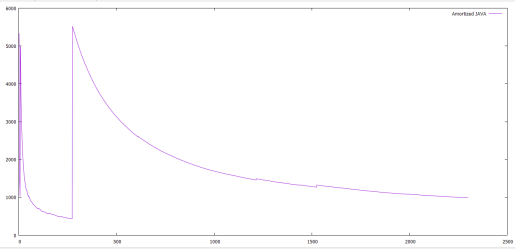




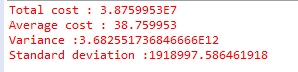
**0,2 :**

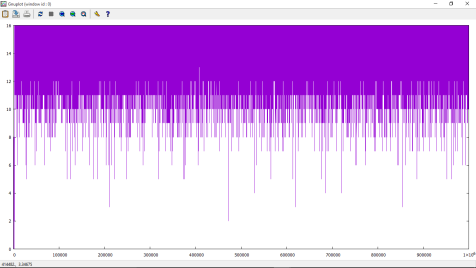
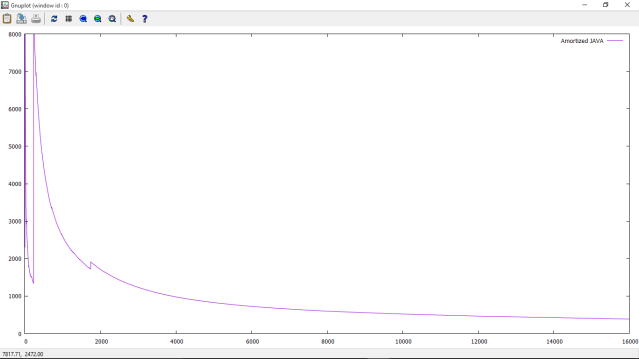
****

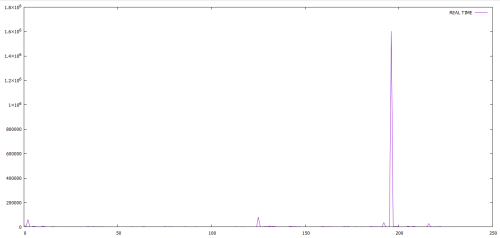
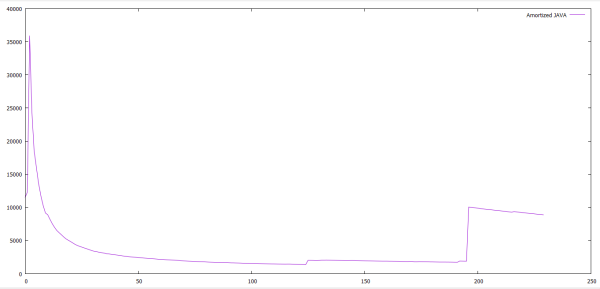
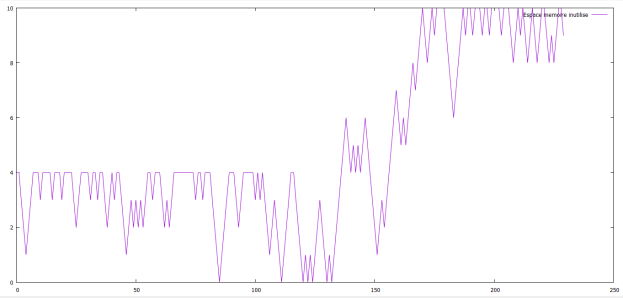




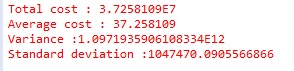
**0,3 :**

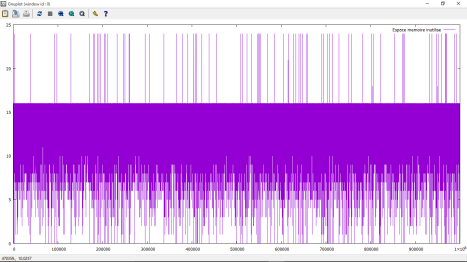
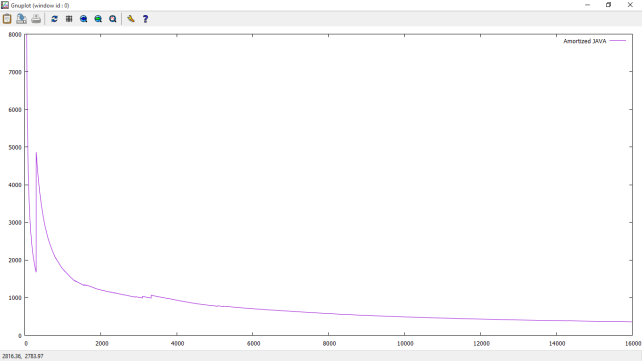




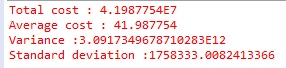


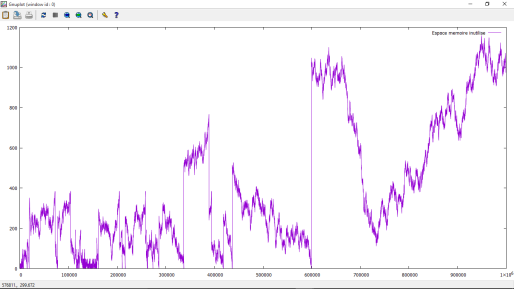
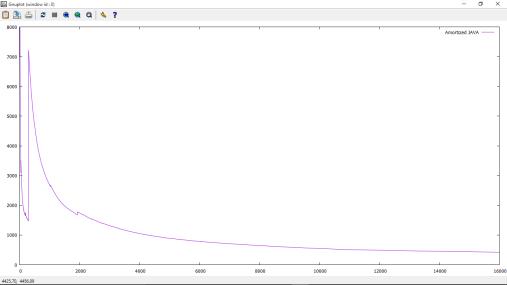
**0,4 :**

****

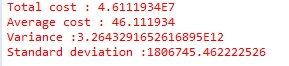


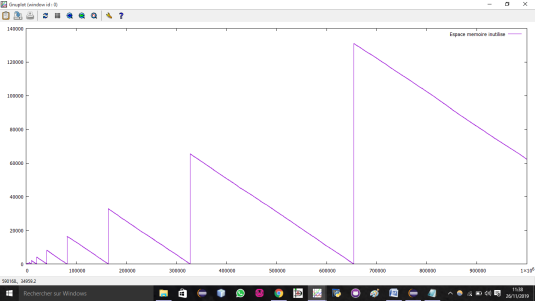
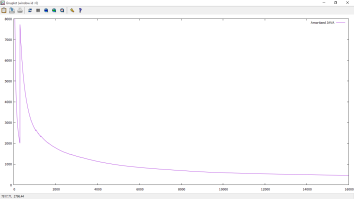
**0,5 :**

****

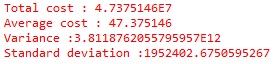


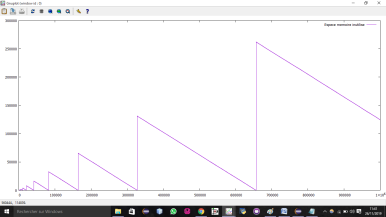
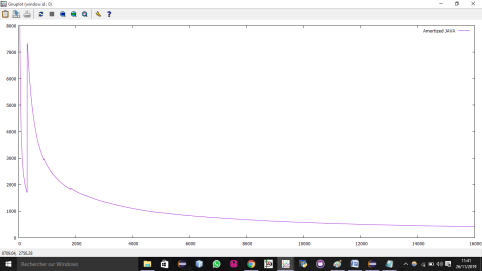
**0,6 :**

****

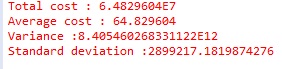


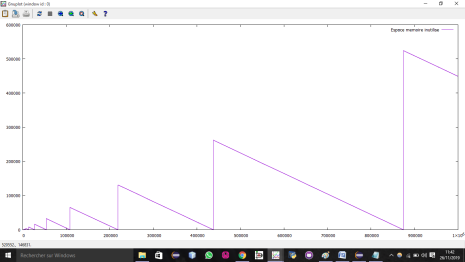
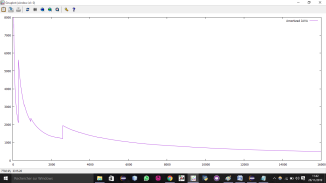
**0,7 :**

****

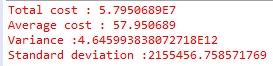


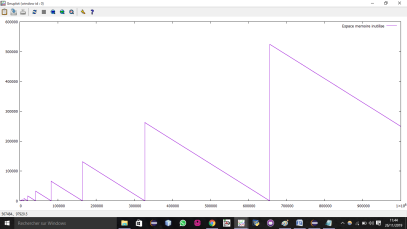
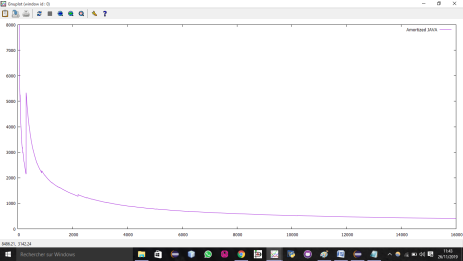
**0,8 :**

****

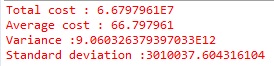


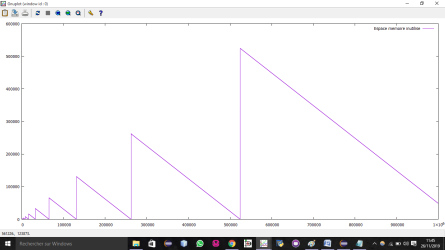
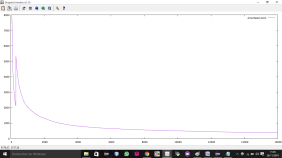
**0,9 :**





**1 :**

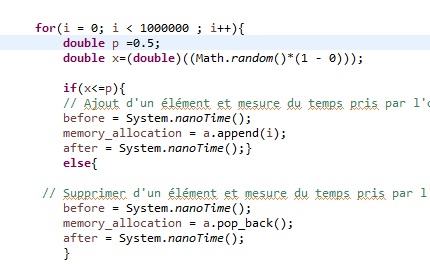
****



Je pense que plus la valeur de p est proche de 1 plus on fait des insertion par rapport ce qui fait que la mémoire inutilisé baisse et le temps inversement pour le p qui se rapproche de 0 donc + de suppressions ce qui fait que mémoire inutilisé augmente

5.

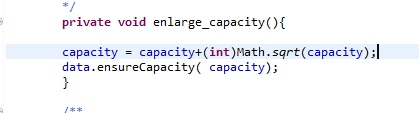
**Quand p=0,5:**



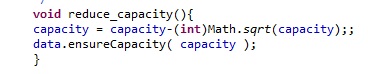
**Quand la table est pleine :**

**C:\Users\VMI\Desktop\reduce.jpg**

**La stratégie lors d’une extension :**



**La stratégie lors d’une contraction :**



**Il faut contracter la table quand une suppression fait passer le remplissage de la table sous les 1/2 :**

**C:\Users\VMI\Desktop\Sans titre.jpg**

Je pense que cette strategie est efficace uniquement si on insere peu d’elementx sinon il va falloir faire beaucoup d’extension donc très couteux en temps, la première stratégie est meilleure.

6.

Je pense que p =0.6 et 0,7 est la meilleure probabilité parce que il y’a plus d’insertions que de suppressions (il y’a assez d’insertions)