Partie B

Question 1)

Fonction	Le pire cas	Le meilleur cas
a)	$O(nlog_2 n)$	$O(1)$
b)	$O(n^2)$	$O(1)$
c)	$O(\infty)$	$O(\infty)$
d)	$O(n * \sqrt{n} * log_3\sqrt{n})$	$O(1)$
e)	$O(n)$	$O(1)$

Question 2)

a)

$$\lim_{n\to\infty} \frac{log^2 n}{logn} = \lim_{n\to\infty} logn * \frac{logn}{logn} = \lim_{n\to\infty} logn = +\infty$$

D'après la limite a) est vraie

b)

$$\lim_{n\to\infty} \frac{(7n^5 + 2n^2 + 7)}{n^3} = \lim_{n\to\infty} \frac{7n^5}{n^3} = \lim_{n\to\infty} 7n^2 = +\infty$$

D'après la limite b) est vraie

c)

$$\lim_{n\to\infty} \frac{n(logn)^2}{n\sqrt{n}} = \lim_{n\to\infty} \left(\frac{logn}{\sqrt{n}}\right) * logn \ or \ la \ \lim_{n\to\infty} \frac{logn}{n^r} = 0$$

$$si \ r \ est \ positif \ par \ consequent \ la \ \lim_{n\to\infty} \left(\frac{logn}{\sqrt{n}}\right) * logn = 0$$

D'après la limite c) est fausse

d)

$$(n^2 - n + 2)^3 \ le \ terme \ qui \ a \ le \ degrès \ le \ plus \ haut \ est \ n^6$$

$$donc \ \lim_{n\to\infty} \frac{(n^2 - n + 2)^3}{n^7} = \lim_{n\to\infty} \frac{n^6}{n^7} = \lim_{n\to\infty} \frac{1}{n} = 0$$

D'après la limite d) est vraie

Question 3)

Dans la méthode récursive notre complexité vaut $O(n^2 logn)$ vu que nous avons utilisé la méthode « sort » de java et sa complexité valait d'après la documentation $nlogn$. Lorsque nous déroulons l'algorithme à chaque itération la complexité était de $2nlogn$. Si notre algorithme est appelé n fois dans la récursivité alors sa complexité va être égale $2n^2 * logn$ d'où $O(n^2 logn)$.

En comparaison avec la méthode itérative nous avons constaté que la complexité était la même. Si nous mettons un boucle « while » qui exécute n fois et que chaque exécution nous avons $2nlogn$ alors au final nous aurons $O(n^2 logn)$.

Donc nous pouvons en conclure que les deux algorithmes s'exécutent en même temps.