Révision examen1

Analyse d'algorithme

1. Notation O dans l'ordre de coût:
    - $O(1)$: prend le même temps, peu importe le nombre d'éléments.
    - $O(1)amortie$: $O(1)$ la plupart du temps
    - $O(\log n)$: divise la donnée en segment à traiter et à ignorer
    - $O(n)$: parcours toute la donnée.
    - $O(n \log n)$: traite toute la donnée mais le traitement prend $\log(n)$ pour chaque élément.
    - $O(n^2)$: deux boucle imbriqués sur l'ensemble de la donnée.
2. Savoir reconnaitre l'ordre l'algorithmique d'un bout de code.

Allocation dynamique

- Identifier ce qui se trouve dans la pile ou sur le tas (heap) à partir d'un bout de code.
- Expliquer le comportement du mot clé new en C++
- Maîtriser l'allocation/dé-allocation d'un pointeur sur un objet (ou type de base) ou sur un tableau.
- Utilisation des pointeurs (déréférence, opérateur [])
- Identifier un problème dans bout de code.
- Régler une fuite de mémoire dans un bout de code.
- Expliquer le comportement d'un segment de code erroné.
- Bonne utilisation des opérateurs -> et .

Vecteur

- Être capable d'implanter un constructeur, copieur, destructeur et opérateur = selon la représentation donnée.
- Expliquer la différence entre la méthode at et l'opérateur []. Être capable de les implanter selon la représentation donnée.
- Être capable d'implanter les méthodes front, back, empty, size et capacity selon la représentation donnée.
- Être capable d'implanter les méthodes $push\_back$, pop_back et $reserve$ selon la représentation donnée.

Ordre algorithmique des méthodes :

- $O(1)$: at, [], constructeur, destructeur, front, back, empty, size, capacity
- $O(1)amortie$: push_back
- $O(n)$: copieur, =, reserve