Phaln .info

Programmation en C++
Survol d'algorithmes de la STL

« Ce métier est une quête infinie,
qui nous rend parfois « schizophrène »
à force de faire et de défaire tout ce qu'on met en place,
est aussi la plus intéressante. »
[Sarah Alem]

p.alluin@phaln.info

09/11/2022

Phaln .info

Introduction aux algorithmes de la STL

- **Des modèles de fonctions**
  - *Résoudre des problèmes récurrents*
  - *Conformes à l'état de l'art*
  - *Performant*
  - *Environs 200 ...*
- **Classification des algorithmes**
  - *Non modifiants*
  - *Modifiants*
  - *Recherche dichotomique*
  - *Partitionnement*
  - *Tri*
  - *Ensemblistes*
  - *Autres...*

Phaln .info

Algorithmes et itérateurs

- **Les algorithmes utilisent des itérateurs**
  - *En interne*
  - *En paramètres d'entrée ou de sortie*
  - *Les itérateurs doivent avoir les propriétés nécessaires (voir pb de sort sur des list<T> ...)*
- **Classification des itérateurs**
  - *Contiguës (contiguous iterator)*
  - *À accès aléatoire (random iterator)*
  - *Bidirectionnels (bidirectionnal iterator)*
  - *Qui avancent (forward iterator)*
  - *D'entrée (input iterator)*
  - *De sortie (output iterator)*
  - *Reculant (reverse iterator)*
  - *Ajout en queue (back insert iterator)*
  - *Ajout en tête (front insert iterator)*

Phaln .info

Programmation en C++
Algorithmes non modifiants

Phaln .info

Algorithmes non modifiants (1)

- **all_of, any_of, none_of**
  - *Savoir si un prédicat (fonction booléenne) est applicable à tous, au moins un, aucun élément*
  - *Type d'itérateurs : input et forward*
- **for_each, for_each_n**
  - *Application d'une fonction à tous, aux n premiers éléments*
  - *Type d'itérateurs : input et forward*
- **count, count_if**
  - *Comptent le nombre d'éléments égaux, satisfaisant, une valeur ou un prédicat*
  - *Type d'itérateurs : input et forward*

Phaln .info

Algorithmes non modifiants (2)

- **mismatch**
  - *Compare les éléments de 2 intervalles et retourne une paire d'itérateurs sur ceux qui diffèrent*
  - *Type d'itérateurs : input et forward*
- **find, find_if, find_if_not**
  - *Recherche le premier élément égal à une valeur, satisfaisant, ne satisfaisant pas un prédicat*
  - *Type d'itérateurs : input et forward*
- **find_end**
  - *Recherche la dernière occurrence d'une séquence d'éléments dans un autre intervalle*
  - *Type d'itérateurs : forward*

Phaln .info

Algorithmes non modifiants (3)

- **find_first_of**
  - *Recherche la première occurrence d'un des éléments d'un intervalle dans un autre intervalle*
  - *Type d'itérateurs : forward et input*
- **adjacent_find**
  - *Recherche la première occurrence de deux éléments consécutifs identiques dans un intervalle*
  - *Type d'itérateurs : forward*
- **search, search_n**
  - *Recherche la première occurrence d'une séquence d'éléments, d'une répétition de n éléments, dans un intervalle*
  - *Type d'itérateurs : forward*

Phaln .info

Programmation en C++
Algorithmes modifiants

Phaln .info

Algorithmes modifiants (1)

- **copy, copy_if, copy_n, copy_backward**
  - *Copie des données (toutes, vérifiant un prédicat, n premières, en ordre inverse) d'un intervalle.*
  - *Type d'itérateurs : input, output, forward, bidirectional*
- **move, move_backward**
  - *Déplacement de données (toutes, en ordre inverse) d'un container vers un autre.*
  - *Type d'itérateurs : input, output, forward*
- **copy, copy_n**
  - *Affecte une valeur à (tous, n premiers) éléments d'un intervalle.*
  - *Type d'itérateurs : output, forward*

Phaln .info

Algorithmes modifiants (2)

- **transform**
  - *Applique une fonction aux éléments d'un intervalle avec le résultat rangé dans un autre intervalle*
  - *Type d'itérateurs : input, output, forward*
- **generate, generate_n**
  - *Affecte le résultat d'un générateur à (tous, n premiers) éléments d'un intervalle.*
  - *Type d'itérateurs : output, forward*
- **remove, remove_if**
  - *Retire des éléments (toutes, vérifiant un prédicat) d'un intervalle et décale les suivants.*
    - ➔ Les derniers éléments ne sont pas supprimés (⇒ erase)
    - ➔ Retourne un itérateur sur la fin logique
  - *Type d'itérateurs : forward*

Phaln .info

Algorithmes modifiants (3)

- **remove_copy, remove_copy_if**
  - *Copie des éléments d'un intervalle vers un autre, sauf ceux vérifiant une valeur, un prédicat*
  - *Type d'itérateurs : input, output, forward*
- **replace, replace_if**
  - *Remplace des éléments (valeur définie, vérifiant un prédicat) d'un intervalle par une valeur déterminée*
  - *Type d'itérateurs : forward*
- **replace_copy, replace_copy_if**
  - *Copie des éléments d'un intervalle vers un autre, en remplaçant ceux vérifiant une valeur, un prédicat par une valeur déterminée*
  - *Type d'itérateurs : input, output, forward*

Phaln .info

Algorithmes modifiants (4)

- **reverse, reverse_copy**
  - *Inverse l'ordre des éléments d'un intervalle, éventuellement lors d'une copie*
  - *Type d'itérateurs : bidirectional*
- **rotate, rotate_copy**
  - *Permutation circulaire des éléments d'un intervalle dans celui d'origine, dans une copie*
  - *Type d'itérateurs : forward, output*
- **unique, unique_copy**
  - *Retire les éléments d'un intervalle, dans celui d'origine, dans une copie, pour qu'il n'y ait plus 2 éléments consécutifs identiques.*
    - ➔ *Les éléments ne sont pas supprimés (⇒ erase)*
    - ➔ *Retourne un itérateur sur la fin logique*
  - *Type d'itérateurs : input, output, forward*

Algorithmes modifiants (5)

- **swap**
  - *Échange les valeurs de deux objets*
  - *Type d'itérateurs : aucun*
- **iter_swap**
  - *Échange les valeurs de deux objets désignés par des itérateurs*
  - *Type d'itérateurs : forward*
- **swap_ranges**
  - *Échange les valeurs des éléments de deux intervalles*
  - *Type d'itérateurs : forward*
- **shuffle**
  - *« Mélange » les valeurs d'un intervalle*
    - ➔ Nécessite un générateur de nombre aléatoire
  - *Type d'itérateurs : aucun*

Programmation en C++

Algorithmes
de partitionnement et de tri

Partitionnement

- **partition**
  - *Déplace les éléments d'un intervalle pour les grouper selon le respect ou non d'un prédicat*
  - *Type d'itérateurs : input, forward*
- **stable_partition**
  - *Idem partition en garantissant le respect de l'ordre initial des deux partitions*
  - *Type d'itérateurs : input, bidirectional*
- **partition_point**
  - *Analyse un intervalle partitionné et retourne un itérateur sur le premier élément de transition*
  - *Type d'itérateurs : forward*

Phaln .info

Tri (1)

- **is_sorted**
  - *Retourne true si les éléments d'un intervalle sont trié (< par défaut)*
  - *Type d'itérateurs : forward*
- **is_sorted_until**
  - *Retourne un itérateur sur le premier élément non trié d'un intervalle (< par défaut)*
  - *Type d'itérateurs : forward*
- **sort**
  - *Trie un intervalle (< par défaut)*
  - *Type d'itérateurs : random*
- **stable_sort**
  - *Idem sort, sans modification d'ordre des doublons*
  - *Type d'itérateurs : random*

Phaln .info

Tri (2)

- **nth_element**
  - *Selon un itérateur quelconque d'un intervalle, déplace les éléments de manière à avoir avant les éléments ≤ et après les éléments >, sans les trier.*
  - *Type d'itérateurs : random*

Phaln .info

Algorithmes modifiants (5)

- **swap**
  - *Échange les valeurs de deux objets*
  - *Type d'itérateurs : aucun*
- **iter_swap**
  - *Échange les valeurs de deux objets désignés par des itérateurs*
  - *Type d'itérateurs : forward*
- **swap_ranges**
  - *Échange les valeurs des éléments de deux intervalles*
  - *Type d'itérateurs : forward*
- **shuffle**
  - *« Mélange » les valeurs d'un intervalle*
    - → Nécessite un générateur de nombre aléatoire
  - *Type d'itérateurs : aucun*