演算法與程式解題實務

Mong-Jen Kao (高孟駿)

Monday 18:30 – 21:20

使用 C++ 的資料容器 (Data Container)

前言

- C++ 的 STL (Standard Template Library) 函式庫提供了多樣化的工具函式/物件, 可以大量簡化程式開發的過程。

  以下我們將約略介紹數種重要的資料容器 (Data Container)

- C++ 的 STL 資料容器包含以下三大類
  - **Sequence Container (序列容器)**
  - Associative Container (關聯容器)
  - Unordered Associative Container (未排序的關聯容器)

前言

- C++ 的 STL 資料容器包含以下三大類
  - **Sequence Container (序列容器)**
    - 顧名思義,
      sequence container 提供了概念上為 list (序列) 的容器

    - 例如：
      - array – 傳統陣列
      - vector – 依實際使用狀況動態配置大小的陣列
      - deque – 雙向佇列 (double-ended queue)

這三個容器,
底層都是以傳統陣列空間實作

使用 array 資料容器

C++ 的 array 容器

- 在概念上, array 容器即為傳統的陣列空間
  - 宣告時, 必須同時決定其大小
  - 裡面的資料, 存放在連續的記憶體空間中

引入 array 需要的標頭檔

- 使用 C++ 的 STL 前, 需要先 include 相對應的 header file

```
#include <array>
```

- 另一個更便利的做法是, 把所有標準函式庫裡的東西全部引入進來

```
#include <bits/stdc++.h>
```

  - 這樣做, 可以把 C 以及 C++ 提供的所有函式庫都一併引入進來

開始使用之前...

可以把它當成慣用語法

- 引入 header files 之後, 放入一行

```
using namespace std;
```

- 若沒有做這個宣告,

  則之後使用 vector 時, 需要再加註是來自 std 命名空間

宣告 array 物件

- 宣告 array 時需要用 < > 加註儲存的資料型態, 以及陣列的大小

```
array<int,10>  my_int_array;
```

宣告了一個長度為 10 個 int 的 array 物件

```
array< array<int,2> ,10>  my_points_array;
```

宣告了一個有 10 個儲存格的 array 物件,
其中每個儲存格為長度為 2 個 int 的 array.

使用 array 物件

```
array<int,10>  my_int_array;
```

- 存取 array 裡的資料 (把它當成傳統陣列使用即可)

  my_int_array[k]

  注意不能超過陣列索引值範圍

- 取得儲存格的 iterator (指標) – 使用 & 運算子

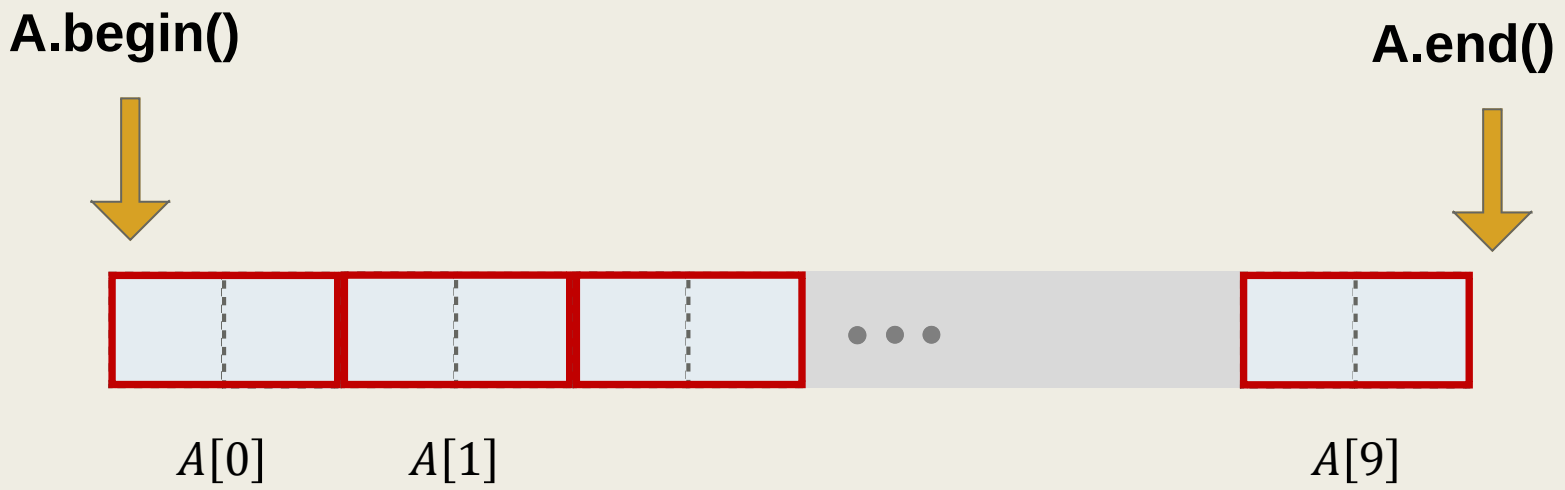
- 相關成員函式
  - size() – 傳回陣列的儲存格個數

    my_int_array.size()

  - begin(), end() – 傳回陣列開頭與結尾的 iterator

STL 資料容器的 Iterator

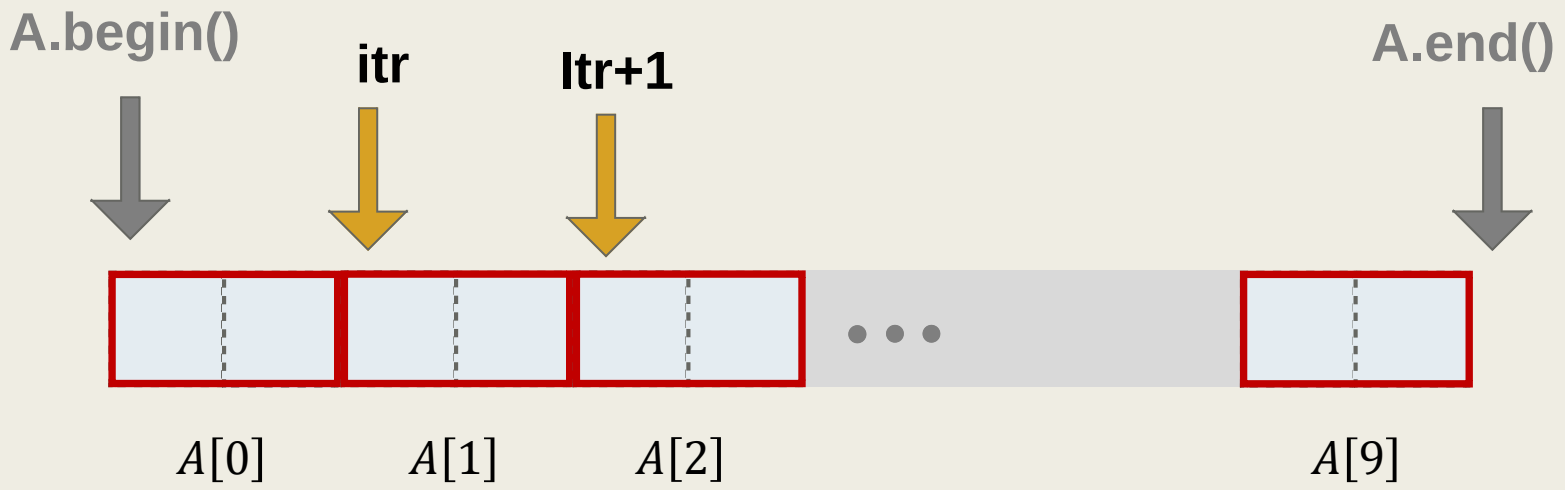
- 先前提到, 可以將 iterator 視為「指向資料容器儲存格」的指標.




```
array< array<int,2>, 10>  A;
```

STL 資料容器的 Iterator

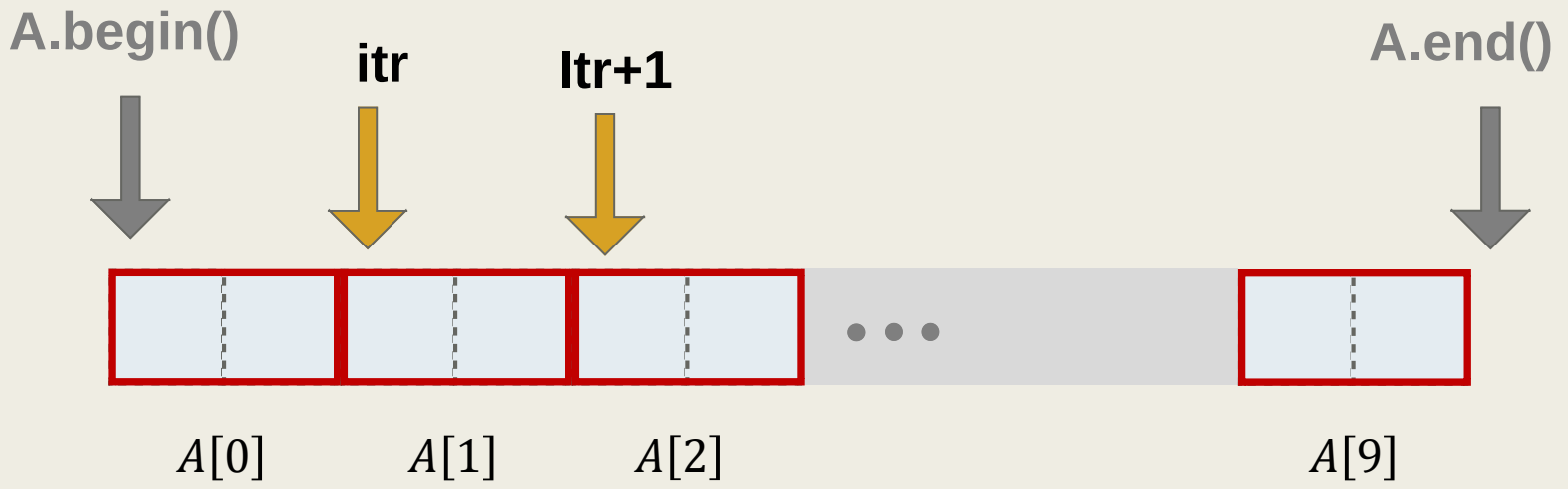
- 先前提到, 可以將 iterator 視為「指向資料容器儲存格」的指標.




```
array< array<int,2>, 10>  A;
```

STL 資料容器的 Iterator

- 對 iterator 使用 * 運算子, 則得到「資料容器儲存格」的 reference.
  - 相對應地, 對儲存格使用 & 運算子, 則得到指向該儲存格的 iterator




```
array< array<int,2>, 10>  A;
```