

Data Structure & Algorithm Review by P.Time

Basic Structure

Primitive class ---> Default class of language

Example int,float,char,double

Secondary class (class) ---> ทุกอย่างนอกจาก primitive

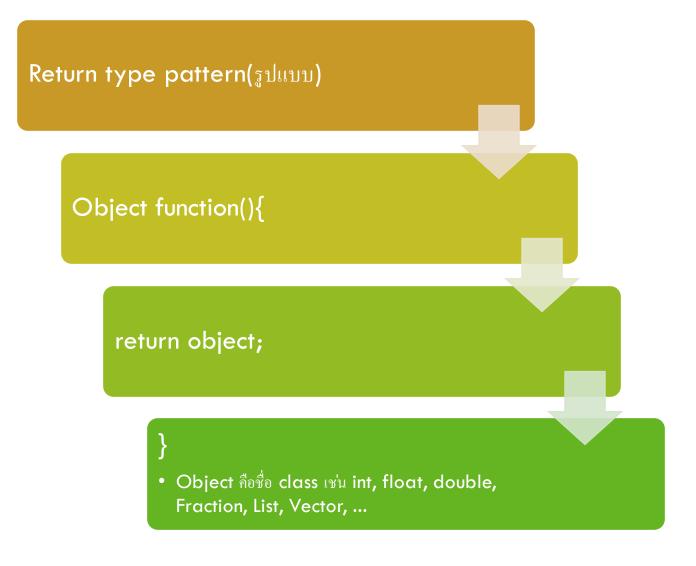
Example list, vector และ class ที่สร้างเอง

Structure

---> Data type (int, float, object ต่างๆ)

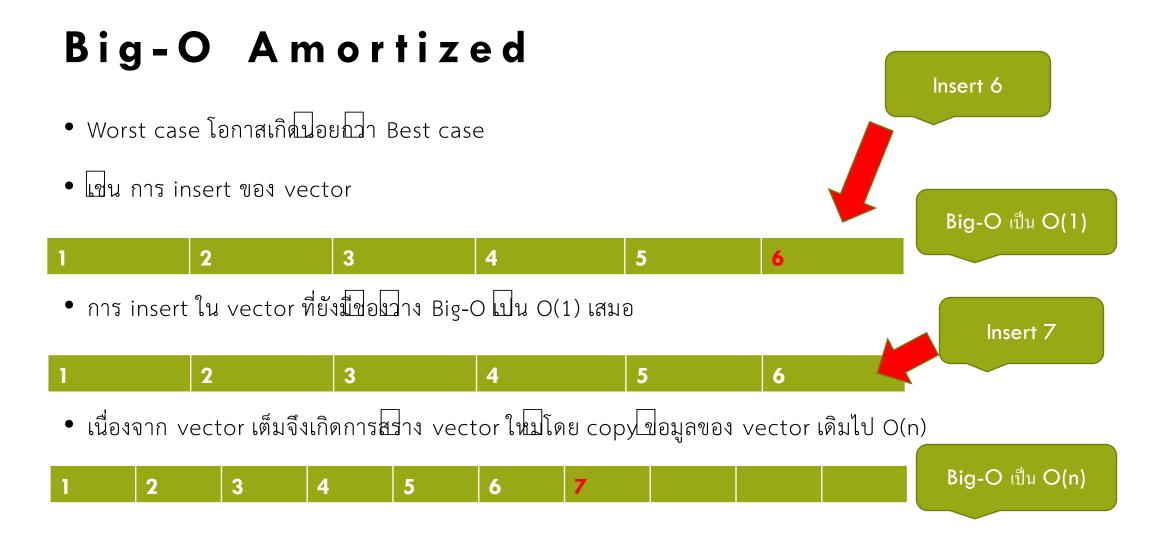
---> Collections (array, vector, list, object ที่เก็บ objects)

functions



Big-O(Run time Complexity)

- โดยปกติ์ฌาพูดถึง Big-O ก็จะพูดถึง worst case
- Average case --> Big theta
- Best case --> Big omega
- Run time Complexity หรือสั้นๆ Big-O คือเวลาที่โชในการทำงานของโปรแกรม
- แบงเป็น O(1) Constant Time, O(logn) Logarithmic Time, O(n) Linear Time
- O(nlogn) Linearithmic Time, O(n^2) Quadratic Time, O(n^3) Cubic Time
- O(2^n) Exponential Time, O(n!) Factorial Time เรียงจากบ่อยสุดไปมากสุด



int *p; การประกาศตัวแปร Pointer ต้องมี * เสมอ

Cout << p; หมายถึง cout << ที่อยู่ของสิ่งที่ pointer p ซื้อยู่

Cout << *p; หมายถึง cout << ค่าของสิ่งที่ P ชื่อยู่

Cout << &p; หมายถึง cout << ที่อยู่ของ pointer p

* เรียกว่า dereferencing operator

& เรียกว่า referencing operator

Pointers

Linked Lists

Singly Linked list

Doubly Linked list

Skip list

Circular list

Singly & Doubly Comparison

Singly linked list (ข้อดี)

- โขาใจงายไรายตอการเขียน
- (Simple)
- โชพื้นที่บอย
- (Require less space)

Doubly linked list (ข้อดี)

- เขาถึง list โด 2 ทาง
- (Two ways to traverse through the list)
- การ delete โซเวลโปอยก์ว่า singly
- (Require less time for node deletion)

Singly & Doubly Comparison

Singly linked list (ข้อเสีย)

- โชเวลา delete นาน
- (take more time to delete node in the list)
- โขาถึง List โดแคทางเดียว
- (one way to traverse through the list)

Doubly linked list (ข้อเสีย)

- โชพื้นที่มาก
- (take more memory space)
- โขาใจไดยากโตองจัดการ pointer หลายตัวมากขึ้น
- (Harder to manage(maintain) the list since it contains one more pointer)

Recursion

- Run time stack complexity
- Direct recursion (Tail recursion, Non-tail Recursion)

Tail recursion -> มีการเรียก recursion ที่การ return ของ function นั้นๆ

- โฆน fibonacci function, factorial function
- Indirect recursion ไม่โดเรียกตัวเองแตเรียก function อื่น แลว function นั้นมาเรียกต่อ

โฆน work() -> bed() -> work() -> bed()

Work เรียก bed และ bed เรียก work ตอเป็นตน

• Easy to read and understand code โคดเขาใจโดเวาย และสวยงาม

Sorting

- Bubble Sort -> Best case: O(n), Worst case: $O(n^{\Lambda}2)$
- Cocktail Sort -> Best case: O(n), Worst case: $O(n^{\Lambda}2)$
- Selection Sort -> Best case: O(n), Worst case: $O(n^{2})$
- Merge Sort -> Best case: O(n), Worst case: O(nlogn)
- Quick Sort -> Best case: O(n), Worst case: O(nlogn)
- Shell Sort -> Best case O(n), Worst case: O(nlogn)

Key questions

- What is the difference between Array, Vector, Linked list
- Explain step by step quick sort of {2, 5, 10, 3, 6, 8, 4, 9, 7, 1}
- Difference between singly and doubly linked list
- Why recursion? When to use it? Why it is bad?