

VÕ TIẾN

Thảo luận kiến thức CNTT trường BK về KHMT(CScience), KTMT(CEngineering)
<https://www.facebook.com/groups/khmt.ktmt.cse.bku>



Cấu Trúc Dữ Liệu và Giải Thuật (DSA)

DSA1 - HK241

Developing List Data Structures and Artificial Neural Networks

Thảo luận kiến thức CNTT trường BK
về KHMT(CScience), KTMT(CEngineering)
<https://www.facebook.com/groups/khmt.ktmt.cse.bku>



Lý Thuyết Bổ Sung ở task 1 (version 1)

1. **Capacity** trong **XArrayList** sẽ được thay đổi nếu không đủ vùng nhớ cấp phát mới

Array Expansion Example

- Ban đầu, array có kích thước **capacity** = 5, với các phần tử: [1, 2, 3, 4, 5].
- Thực hiện lệnh **array.add(6)**. Do vùng nhớ hiện tại không đủ, ta cần cấp phát thêm.
- Quy tắc cấp phát thêm: **capacity = capacity * 2 + 1**.
- Sau khi cấp phát thêm, array có giá trị: [1, 2, 3, 4, 5, 6, 0, 0, 0, 0, 0].

2. **Move** các đối tượng khi gọi **removeAt** trong **XArrayList**

Array Expansion Example

- Ban đầu, array có kích thước **capacity** = 5, với các phần tử: [1, 2, 3, 4, 5].
- Thực hiện lệnh **array.removeAt(2)**. Phần tử tại vị trí thứ 2 (giá trị 3) sẽ bị xóa.
- Các phần tử sau vị trí 2 sẽ dồn lên đầu để lấp chỗ trống:
- Kết quả sau khi xóa: [1, 2, 4, 5, 0].
- Lưu ý: Phần tử cuối cùng được thay thế bằng 0 để thể hiện giá trị không hợp lệ.

3. **Point Function itemEqual**

- Trường hợp **Type** của **data** là kiểu nguyên thủy và **itemEqual** = 0

Array Expansion Example

- Ban đầu chứa các phần tử: [1, 2, 3, 4, 5].
- Khi thực hiện lệnh **array.indexOf(2)**, hệ thống sẽ tìm giá trị 2 trong mảng.
- Để so sánh giá trị, hệ thống sử dụng toán tử **==**. Nếu giá trị 2 được tìm thấy, phương thức **indexOf** sẽ trả về chỉ số của phần tử đó.
- Nếu giá trị 2 không có trong mảng, phương thức **indexOf** sẽ trả về -1.

- Trường hợp **Type** của **data** là kiểu nguyên thủy và **itemEqual** = **foo(int a, int b) {return abs(a) == abs(b);}**

Array Expansion Example

- Ban đầu, mảng chứa các phần tử: [1, 2, -3, 4, 5].
- Hàm so sánh được sử dụng là **itemEqual(int a, int b) { return abs(a) == abs(b); }**. Hàm này so sánh giá trị tuyệt đối của hai số.
- Khi thực hiện lệnh **array.indexOf(3)**, hệ thống sẽ tìm giá trị 3 trong mảng, nhưng so sánh sẽ dựa trên giá trị tuyệt đối. Vì giá trị tuyệt đối của -3 là 3, nên **itemEqual** sẽ coi -3 là phần tử bằng với 3.
- Phương thức **indexOf** sẽ trả về chỉ số của phần tử có giá trị tuyệt đối bằng 3, trong trường hợp này là 2 (vì -3 có chỉ số là 2 trong mảng).
- Nếu không có phần tử nào có giá trị tuyệt đối bằng 3, phương thức **indexOf** sẽ trả về -1.

- Trường hợp **Type** của **data** là kiểu đối tượng và có **overload operator==**



Array Expansion Example

- Giả sử bạn có một mảng chứa các đối tượng kiểu `Point` với các giá trị: `[Point(1, 2, 3), Point(4, 5, 6), Point(7, 8, 9)]`.
- Đối tượng `Point` đã overload toán tử `==` để so sánh các đối tượng dựa trên các thuộc tính của chúng (ví dụ: so sánh từng tọa độ `x`, `y`, `z`).
- Khi thực hiện lệnh `array.indexOf(Point(4, 5, 6))`, hệ thống sẽ tìm đối tượng `Point(4, 5, 6)` trong mảng.
- Để so sánh đối tượng, hệ thống sẽ sử dụng toán tử `==` đã được overload. Nếu đối tượng `Point(4, 5, 6)` được tìm thấy trong mảng, phương thức `indexOf` sẽ trả về chỉ số của đối tượng đó.
- Nếu đối tượng không có trong mảng, phương thức `indexOf` sẽ trả về `-1`.

- Trường hợp **Type** của **data** là kiểu đối tượng và **itemEqual = pointEQ_X(Point a, Point b) {return a.x == b.x;}**

Array Expansion Example

- Giả sử bạn có một mảng chứa các đối tượng kiểu `Point` với các giá trị: `[Point(1, 2, 3), Point(4, 5, 6), Point(7, 8, 9)]`.
- Lớp `Point` có một hàm so sánh tùy chỉnh `pointEQ_X` để so sánh các đối tượng dựa trên thuộc tính `x` của chúng. Hàm này có dạng:

```
bool pointEQ_X(Point a, Point b) { return a.x == b.x; }
```
- Khi thực hiện lệnh `array.indexOf(Point(4, 5, 6), pointEQ_X)`, hệ thống sẽ tìm đối tượng `Point(4, 5, 6)` trong mảng.
- Để so sánh đối tượng, hệ thống sẽ sử dụng hàm `pointEQ_X`, thay vì toán tử `==`. Hàm này chỉ so sánh thuộc tính `x` của các đối tượng.
- Nếu đối tượng `Point` nào có thuộc tính `x` trùng khớp với `4` được tìm thấy trong mảng, phương thức `indexOf` sẽ trả về chỉ số của đối tượng đó.
- Nếu không có đối tượng nào thỏa mãn điều kiện so sánh, phương thức `indexOf` sẽ trả về `-1`.

4. **Thiết kế một danh sách liên kết đôi (Doubly Linked List) với hai node giả (dummy nodes)** vì iterator của thầy đang hiện thực theo hướng đó nếu mấy bạn không muốn thì đổi code iterator của thầy

```

1 DLinkedList() {
2     // Tạo hai node giả head và tail
3     head = new Node(0);
4     tail = new Node(0);
5
6     // Kết nối head và tail
7     head->next = tail;
8     tail->prev = head;
9 }

```