**Kết quả từ AI:**

1. So sánh Array và Linked List

|  |  |  |
| --- | --- | --- |
| Tiêu chí | Array (Mảng) | Linked List (Danh sách liên kết) |
| Truy cập (Access) | O(1) – có thể truy cập trực tiếp phần tử qua chỉ số arr[i] | O(n) – phải duyệt từ đầu danh sách đến vị trí cần tìm |
| Chèn (Insert) | O(n) – nếu chèn ở giữa phải dời các phần tử sau đó sang phải | O(1) nếu chèn ở đầu/cuối (nếu biết con trỏ), O(n) nếu chèn ở giữa |
| Xóa (Delete) | O(n) – phải dồn các phần tử sau đó sang trái | O(1) nếu xóa ở đầu/cuối (nếu biết con trỏ), O(n) nếu xóa ở giữa |
| Bộ nhớ (Memory) | Liên tục, dễ quản lý, nhưng có thể lãng phí do cấp phát thừa hoặc phải resize | Phân tán trong bộ nhớ, cần thêm bộ nhớ lưu con trỏ next cho mỗi node |
| Tình huống áp dụng | Khi cần **truy cập nhanh** bằng chỉ số, số phần tử **cố định/ít thay đổi** | Khi cần **chèn/xóa nhiều**, số phần tử thay đổi linh hoạt |

1. Code minh họa thao tác chèn phân tử
   1. Chèn vào Array

Ví dụ chèn số 99 vào vị trí pos trong mảng:

public class ArrayInsertDemo {

public static void main(String[] args) {

int[] arr = new int[10];

int n = 5; // số phần tử hiện có

arr[0] = 10; arr[1] = 20; arr[2] = 30; arr[3] = 40; arr[4] = 50;

int pos = 2; // chèn vào vị trí thứ 2 (index 2)

int value = 99;

// Dịch các phần tử sang phải

for (int i = n; i > pos; i--) {

arr[i] = arr[i - 1];

}

arr[pos] = value;

n++; // tăng số lượng phần tử

System.out.println("Array sau khi chèn:");

for (int i = 0; i < n; i++) {

System.out.print(arr[i] + " ");

}

}

}

* 1. Chèn vào **Linked List**

Ví dụ chèn số 99 vào vị trí pos trong danh sách liên kết đơn:

class Node {

int data;

Node next;

Node(int data) {

this.data = data;

this.next = null;

}

}

public class LinkedListInsertDemo {

Node head;

// Hàm chèn vào vị trí bất kỳ

public void insertAt(int pos, int value) {

Node newNode = new Node(value);

if (pos == 0) { // chèn đầu

newNode.next = head;

head = newNode;

return;

}

Node current = head;

for (int i = 0; current != null && i < pos - 1; i++) {

current = current.next;

}

if (current == null) {

System.out.println("Vị trí không hợp lệ!");

return;

}

newNode.next = current.next;

current.next = newNode;

}

public void display() {

Node temp = head;

while (temp != null) {

System.out.print(temp.data + " ");

temp = temp.next;

}

System.out.println();

}

public static void main(String[] args) {

LinkedListInsertDemo list = new LinkedListInsertDemo();

list.insertAt(0, 10);

list.insertAt(1, 20);

list.insertAt(2, 30);

list.insertAt(3, 40);

list.insertAt(2, 99); // chèn 99 vào vị trí index 2

System.out.println("Linked List sau khi chèn:");

list.display();

}

}

* Vì vậy nên không cần chỉnh prompt nữa.