**Bài Toán:**

Bạn có một kho hàng chứa danh sách mã số sản phẩm được sắp xếp theo thứ tự tăng dần. Bạn cần tìm một sản phẩm cụ thể bằng mã số của nó. Thuật toán tìm kiếm nhị phân là một phương pháp hiệu quả để thực hiện việc này trong một mảng đã được sắp xếp.

 **Danh sách mã sản phẩm (productCodes):**

* Mảng productCodes chứa danh sách mã số sản phẩm được sắp xếp theo thứ tự tăng dần. Đây là dữ liệu kho hàng bạn đang làm việc.

 **Nhập mã sản phẩm từ người dùng:**

* Console.Write("Nhập mã số sản phẩm cần tìm: "); yêu cầu người dùng nhập mã sản phẩm mà họ muốn tìm.
* int.TryParse(Console.ReadLine(), out targetCode) kiểm tra xem giá trị nhập vào có phải là số nguyên hợp lệ không. Nếu không, chương trình sẽ yêu cầu nhập lại cho đến khi nhận được giá trị hợp lệ.

 **Gọi hàm BinarySearch:**

* Hàm BinarySearch nhận mảng productCodes và mã sản phẩm cần tìm (targetCode) để thực hiện tìm kiếm nhị phân.

 **Hàm BinarySearch:**

* **Khởi tạo:** left và right xác định phạm vi tìm kiếm ban đầu.
* **Vòng lặp while:** Tiếp tục cho đến khi phạm vi tìm kiếm không còn hợp lệ (left > right).
  + **Tính chỉ số giữa (mid):** Tính chỉ số giữa của phạm vi hiện tại.
  + **So sánh:**
    - Nếu phần tử giữa (array[mid]) bằng target, trả về chỉ số mid.
    - Nếu target nhỏ hơn phần tử giữa, thu hẹp phạm vi tìm kiếm sang nửa bên trái (right = mid - 1).
    - Nếu target lớn hơn phần tử giữa, thu hẹp phạm vi tìm kiếm sang nửa bên phải (left = mid + 1).
* **Kết quả không tìm thấy:** Nếu vòng lặp kết thúc mà không tìm thấy giá trị, trả về -1.

 **Hiển thị kết quả:**

* Nếu result khác -1, thông báo mã sản phẩm được tìm thấy và vị trí của nó trong mảng.
* Nếu result là -1, thông báo rằng sản phẩm không tồn tại trong kho.

 **Chờ người dùng nhấn phím Enter:**

* Console.ReadLine() được sử dụng để giữ cửa sổ console mở cho đến khi người dùng nhấn Enter, giúp bạn có thời gian đọc kết quả trước khi đóng cửa sổ.

### **Chi Tiết Hàm BinarySearch**

Hàm BinarySearch thực hiện tìm kiếm nhị phân trên mảng đã được sắp xếp. Đây là cách mà hàm này hoạt động:

1. **Khởi tạo các chỉ số:**
   * int left = 0;: Chỉ số bắt đầu của mảng.
   * int right = array.Length - 1;: Chỉ số kết thúc của mảng.
2. **Vòng lặp while:**
   * Điều kiện lặp left <= right: Vòng lặp tiếp tục cho đến khi phạm vi tìm kiếm không còn hợp lệ. Nếu left lớn hơn right, có nghĩa là phần tử cần tìm không nằm trong mảng.
3. **Tính chỉ số giữa (mid):**
   * int mid = left + (right - left) / 2;: Tính chỉ số giữa của phạm vi hiện tại. Sử dụng (right - left) / 2 thay vì (left + right) / 2 để tránh tràn số nguyên trong một số trường hợp (mặc dù trong các ngôn ngữ hiện đại, việc này ít xảy ra).
4. **So sánh phần tử giữa với giá trị cần tìm (target):**
   * **Tìm thấy phần tử:** Nếu array[mid] == target, trả về mid, chỉ số của phần tử tìm thấy.
   * **Nhỏ hơn phần tử giữa:** Nếu target < array[mid], giá trị cần tìm nằm trong nửa bên trái của mảng. Cập nhật right thành mid - 1.
   * **Lớn hơn phần tử giữa:** Nếu target > array[mid], giá trị cần tìm nằm trong nửa bên phải của mảng. Cập nhật left thành mid + 1.
5. **Kết thúc vòng lặp:**
   * Nếu vòng lặp kết thúc mà không tìm thấy giá trị (left > right), trả về -1 để báo rằng phần tử không có trong mảng.

### **Ví Dụ Minh Họa**

Giả sử bạn có mảng productCodes = {10000, 10050, 10100, 10123, 10200, 10300, 15000, 20000} và bạn muốn tìm mã sản phẩm 10200.

1. **Khởi tạo:**
   * left = 0, right = 7.
2. **Lần đầu vòng lặp:**
   * mid = 3, phần tử giữa là 10123.
   * 10200 > 10123, do đó, tìm trong nửa bên phải (left = 4, right = 7).
3. **Lần thứ hai vòng lặp:**
   * mid = 5, phần tử giữa là 10300.
   * 10200 < 10300, do đó, tìm trong nửa bên trái (left = 4, right = 4).
4. **Lần thứ ba vòng lặp:**
   * mid = 4, phần tử giữa là 10200.
   * 10200 == 10200, tìm thấy phần tử tại chỉ số 4.
5. **Kết quả:**
   * Phần tử 10200 được tìm thấy tại vị trí thứ 5 trong mảng (chỉ số 4).

### Ghi Chú Thêm

* **Hiệu suất:** Tìm kiếm nhị phân có độ phức tạp thời gian O(log n), nơi n là số lượng phần tử trong mảng. Điều này làm cho nó rất hiệu quả cho các mảng lớn.
* **Yêu cầu:** Mảng phải được sắp xếp trước khi thực hiện tìm kiếm nhị phân. Nếu không, kết quả có thể không chính xác.

Đoạn mã này giúp bạn tìm sản phẩm trong kho hàng một cách nhanh chóng và hiệu quả nhờ vào thuật toán chia để trị, cụ thể là tìm kiếm nhị phân