

# BÁO CÁO TIỂU LUẬN MÔN THUẬT TOÁN VÀ ỨNG DỤNG

ĐỀ TÀI: XÂY DỰNG CHƯƠNG TRÌNH MINH HỌA CÁC  
THUẬT TOÁN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Giáo viên hướng dẫn: Nguyễn Đình Hiền

Sinh viên thực hiện:

Hoàng Minh Tài	6051071102
Nguyễn Hoàng Hiệp	6051071147
Cao Lâm Bảo Khanh	6051071056
Trương Được	6051071033
Nguyễn Hữu Đại	6051071023



Lớp: CNTT K60

# NỘI DUNG CHÍNH

1

Lý do chọn đề tài

2

Phạm vi đề tài và  
kết quả đạt được

3

Giới thiệu về các  
thuật toán

4

Demo

# 1. Lý do chọn đề tài

Cấu trúc dữ liệu và Giải thuật là môn học quan trọng trong lập trình. Tất cả sinh viên khi theo học công nghệ thông tin sẽ được học cấu trúc dữ liệu và giải thuật. Các vấn đề cần giải quyết trong lập trình đều sử dụng các giải thuật. Những giải thuật cơ bản bao gồm sắp xếp, tìm kiếm, danh sách liên kết, cây nhị phân tìm kiếm... Vì vậy việc minh họa sẽ giúp các sinh viên cũng như ai đang tìm hiểu sẽ hiểu rõ hơn.

## 2. Phạm vi đề tài và kết quả đạt được

Xây dựng thành công chương trình minh họa từng bước các thuật toán trong môn cấu trúc dữ liệu và giải thuật.

## 2. Phạm vi đề tài và kết quả đạt được

Kết quả đạt được:

- Xây dựng thành công được chương trình minh họa từng bước các thuật toán trong môn cấu trúc dữ liệu và giải thuật.

Ưu điểm:

- Dễ nhìn, sinh động
- Phù hợp với những người bắt đầu học thuật toán

Nhược điểm:

- Hạn chế về số lượng các thuật toán
- Bố cục chưa đẹp

## 3. Giới thiệu về các thuật toán



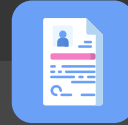
### 3.1 Bubble sort

Sắp xếp nổi bọt



### 3.2 Selection sort

Sắp xếp chọn



### 3.3 Insert sort

Sắp xếp chèn



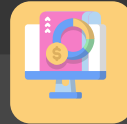
### 3.4 Merge sort

Sắp xếp trộn



### 3.5 Quick sort

Sắp xếp nhanh



### 3.6 Link List

Danh sách liên kết

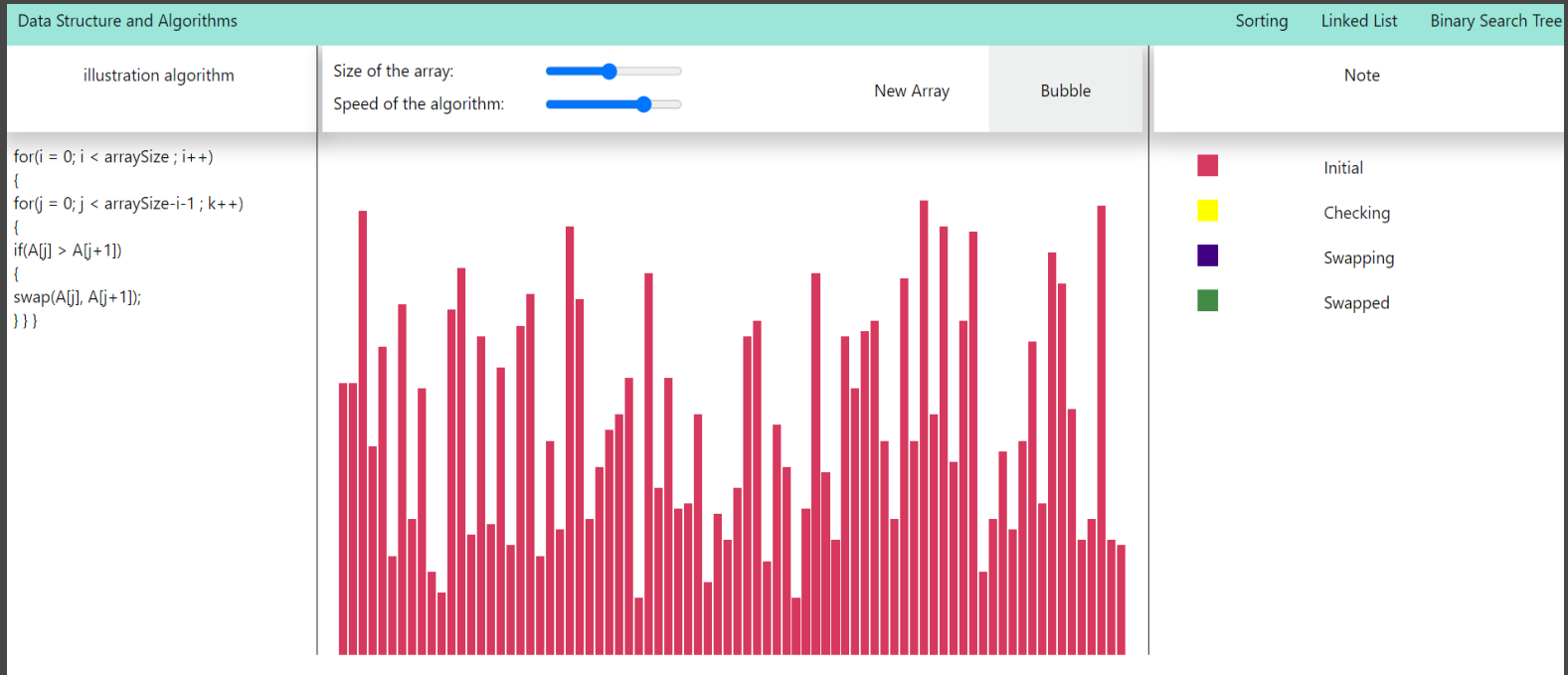


### 3.7 Binary search tree

Cây nhị phân tìm kiếm

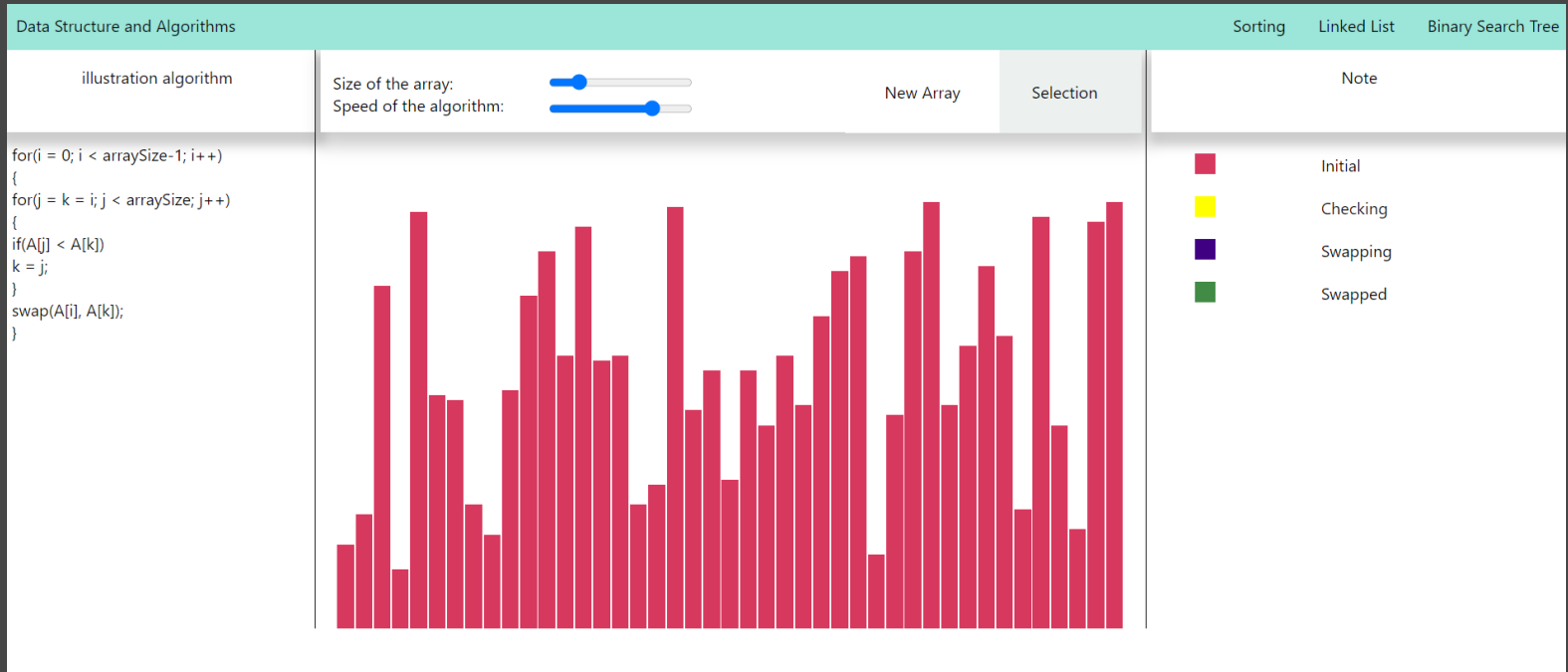
# 3. Giới thiệu về các thuật toán

## 3.1 Giao diện chính của thuật toán Bubble sort



### 3. Giới thiệu về các thuật toán

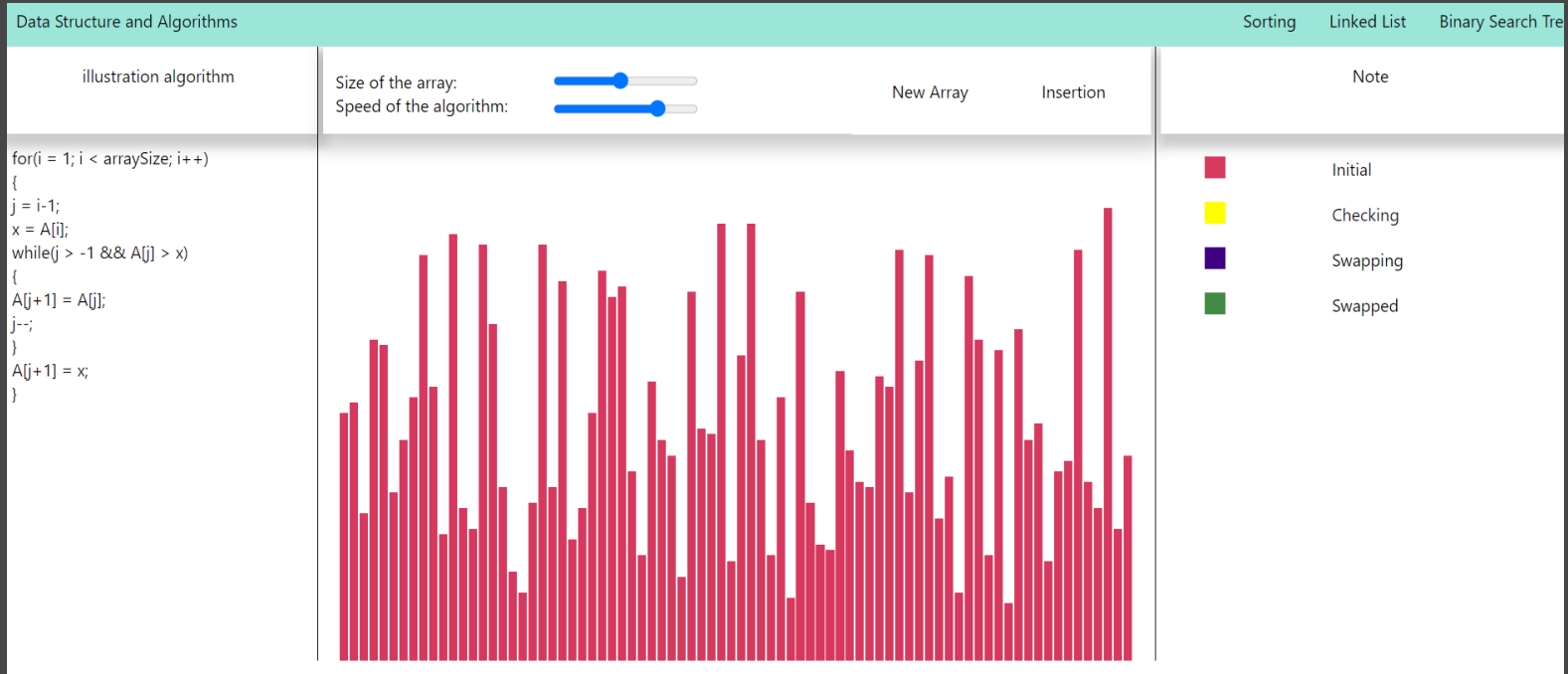
#### 3.2 Giao diện chính của thuật toán Selection sort





## 3. Giới thiệu về các thuật toán

### 3.3 Giao diện chính của thuật toán Insert sort



# 3. Giới thiệu về các thuật toán

## 3.4 Giao diện chính của thuật toán Merge sort



Data Structure and Algorithms

SortingLinked ListBinary Search Tree

illustration algorithm

Size of the array:   
Speed of the algorithm:

New ArrayMerge

```
MergeSort(B[], start, mid, end){  
  p = start, q = mid+1;  
  A[size], k = 0;  
  for(i = start; i <= end; i++)  
  {  
    if(p > mid)  
      A[k++] = B[q++];  
    else if(q > end)  
      A[k++] = B[p++];  
    else if(B[p] < B[q])  
      A[k++] = B[p++];  
    else  
      A[k++] = B[q++];  
  }  
  for(t = 0; t < k; t++)  
    B[start++] = A[t];  
  MergePartition(start, end){  
    if(start < end)  
    {  
      Mid=(start+end)/2;  
      MergePartition(start, mid);  
      MergePartition(mid+1, end);  
      MergeSort(Array, start, mid, end);  
    }  
  }  
}
```

Initial  
Checking  
Swapping  
Swapped

# 3. Giới thiệu về các thuật toán

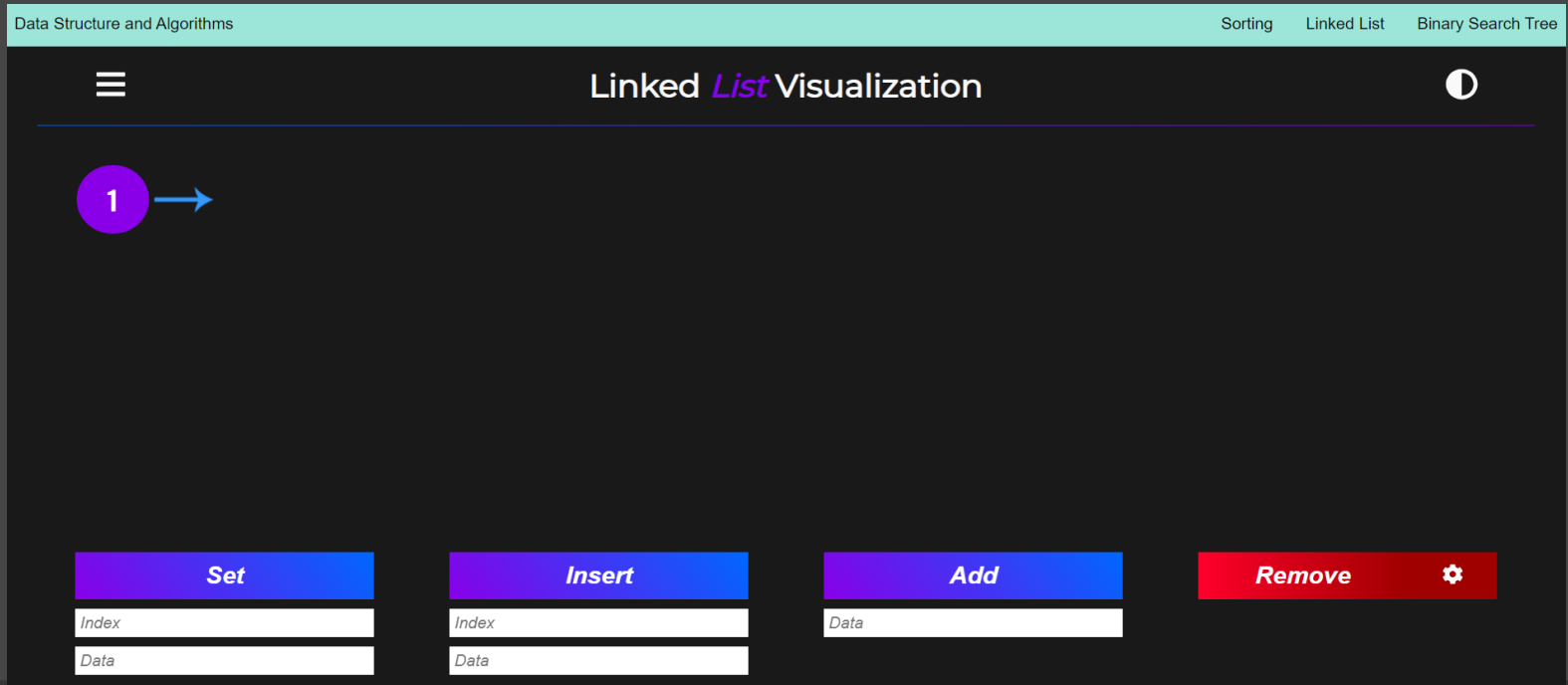
## 3.5 Giao diện chính của thuật toán Quick sort



Data Structure and Algorithms		Sorting	Linked List	Binary Search Tree
illustration algorithm	Size of the array: Speed of the algorithm:	New Array	Quick	Note
<pre>Partition(Arr[ ],start,end) { i=start+1; pivot=Arr[start]; for(j=start+1;j&lt;=end;j++) { if(Arr[j] &lt; pivot) { swap(Arr[i], Arr[j]); i+=1; } } swap(Arr[start], Arr[i-1]); return (i-1); } QuickSort(start, end) { if(start &lt; end) { pivot=Partition(Arr[], start, end) QuickSort(start, pivot - 1); QuickSort(Pivot+1, end); } }</pre>				<ul style="list-style-type: none"><li><span style="color: red;">■</span> Initial</li><li><span style="color: yellow;">■</span> Checking</li><li><span style="color: purple;">■</span> Swapping</li><li><span style="color: green;">■</span> Swapped</li></ul>

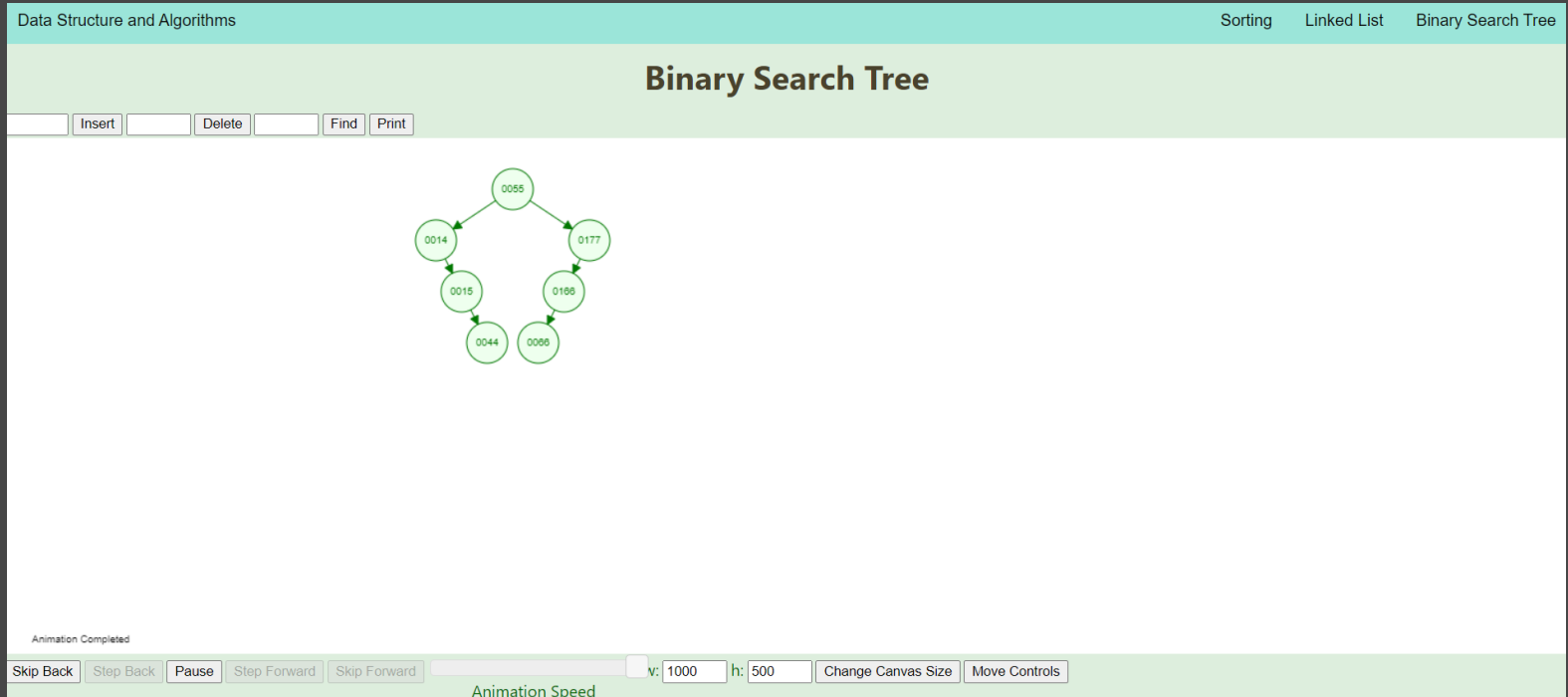
## 3. Giới thiệu về các thuật toán

### 3.6 Giao diện chính của thuật toán Linked List



# 3. Giới thiệu về các thuật toán

## 3.7 Giao diện chính của thuật toán Binary Search Tree



4

# Demo Chương Trình

