Họ và Tên: Phan Văn Bằng

Lớp: 20IT1

DATA STRUCTURE

Assignment 8 – Searching and Sorting

**Problem 1.** Show the steps that Linear search finds a given value x = 4 in an array A = {10, 8, 2, 7, 3, 4, 9, 1, 6, 5}.

Tìm x = 4;

A = {**10,** 8, 2, 7, 3, 4, 9, 1, 6, 5}. I = 0

A = {10, **8,** 2, 7, 3, 4, 9, 1, 6, 5}. I = 1

A = {10, 8, **2,** 7, 3, 4, 9, 1, 6, 5}. I = 2

A = {10, 8, 2, **7,** 3, 4, 9, 1, 6, 5}. I = 3

A = {10, 8, 2, 7, **3,** 4, 9, 1, 6, 5}. I = 4

A = {10, 8, 2, 7, 3, 4, 9, 1, 6, 5}. I = 5

* Tìm ra x = 4 tại chỉ mục i = 5;

**Problem 2.** Show the steps that Binary search finds a given value x = 4 in a sorted array A = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}.

Tìm x = 4;

A = {**1,** 2, 3, 4, 5, 6, 7, 8, 9, 10} 1<4

A = {1, **2**, 3, 4, 5, 6, 7, 8, 9, 10} 2<4

A = {1, 2, **3**, 4, 5, 6, 7, 8, 9, 10} 3<4

A = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} 4=4

Tìm ra

**Problem 3**. Given an array A = {10, 8, 2, 7, 3, 4, 9, 1, 6, 5}. Visualize how the array is sorted step by step in case of:

***a) Insertion sort***

* Hình dung cách sắp xếp mảng từng bước trong trường hợp:

A = {**10,** 8, 2, 7, 3, 4, 9, 1, 6, 5}

A = {**8, 10,** 2, 7, 3, 4, 9, 1, 6, 5}

A = {**2, 8, 10**, 7, 3, 4, 9, 1, 6, 5}

A = {**2, 7, 8, 10,** 3, 4, 9, 1, 6, 5}

A = {**2, 3, 7, 8, 10,** 4, 9, 1, 6, 5}

A = {**2, 3, 4, 7, 8, 10,** 9, 1, 6, 5}

A = {**2, 3, 4, 7, 8, 9, 10,** 1, 6, 5}

A = {**1, 2, 3, 4, 7, 8, 9, 10,** 6, 5}

A = {**1, 2, 3, 4, 6, 7, 8, 9, 10**, 5}

A = {**1, 2, 3, 4, 6, 5, 7, 8, 9, 10**}

* Trong Insertion sort, ta Lặp lại từ arr [1] đến arr [n] trên mảng. So sánh phần tử hiện tại với phần tử tiền nhiệm của nó. Nếu phần tử hiện tại nhỏ hơn phần tử trước của nó, hãy so sánh nó với các phần tử trước đó. Di chuyển các phần tử lớn hơn lên thêm một vị trí để tạo khoảng trống cho phần tử được hoán đổi.
* Hàm:

void insertionSort (int a[], int size) {

int temp, pass, k;

for (pass = 1; pass <= size - 1; pass++) {

for ( k = pass; k >= 1; k--) {

if (a[k] < a[k-1]) {

temp = a[k];

a[k] = a[k-1];

a[k-1] = temp;

}

}

}

return;

}

***b) Selection sort.***

* Hình dung cách sắp xếp mảng từng bước trong trường hợp:

A = {10, 8, 2, 7, 3, 4, 9, 1, 6, 5}

A = {**1**, 8, 2, 7, 3, 4, 9, 10, 6, 5}

A = {**1, 2,** 8, 7, 3, 4, 9, 10, 6, 5}

A = {**1, 2, 3**, 7, 8, 4, 9, 10, 6, 5}

A = {**1, 2, 3, 4,** 8, 7, 9, 10, 6, 5}

A = {**1, 2, 3, 4, 5,** 7, 9, 10, 6, 8}

A = {**1, 2, 3, 4, 5, 6**, 9, 10, 7, 8}

A = {**1, 2, 3, 4, 5, 6, 7,** 10, 9, 8}

A = {**1, 2, 3, 4, 5, 6, 7, 8,** 9, 10}

A = {**1, 2, 3, 4, 5, 6, 7, 8, 9, 10**}

* Trong Selection sort, ta bắt đầu với vị trí đầu tiên trong mảng, sau đó tìm giá trị nhỏ nhất trong lần duyệt đầu tiên, và hoán đổi giá trị ở vị trí đầu tiên trong mảng với giá trị ở vị trí có giá trị nhỏ nhất mới tìm thấy. Sau đó ta lại lấy từng vị trí tiếp theo trong mảng, tìm giá trị nhỏ nhất trong mảng còn lại và hoán đổi vị trí.
* Hàm:

void selectionSort (int a[], int size) {

int temp, pass, k, min, minIndex;

for (pass = 0; pass < size - 1; pass++) {

min = a[pass]; minIndex = pass;

for (k = pass + 1; k < size; k++) {

if (a[k] < min) {

min = a[k];

minIndex = k;

}

}

temp = a[pass];

a[pass] = a[minIndex];

a[minIndex] = temp;

}

return;

}

***c) Bubble sort*.**

* Hình dung cách sắp xếp mảng từng bước trong trường hợp:

A = {10, 8, 2, 7, 3, 4, 9, 1, 6, 5}

A = {**8, 10**, 2, 7, 3, 4, 9, 1, 6, 5}

A = {8, **2, 10**, 7, 3, 4, 9, 1, 6, 5}

A = {8, 2, **7, 10,** 3, 4, 9, 1, 6, 5}

A = {8, 2, 7, **3, 10,** 4, 9, 1, 6, 5}

A = {8, 2, 7, 3, **4, 10,** 9, 1, 6, 5}

A = {8, 2, 7, 3, 4, **9, 10**, 1, 6, 5}

A = {8, 2, 7, 3, 4, 9, **1, 10**, 6, 5}

A = {8, 2, 7, 3, 4, 9, 1, **6, 10**, 5}

A = {8, 2, 7, 3, 4, 9, 1, 6, **5, 10**}

A = {**2, 8**, 7, 3, 4, 9, 1, 6, 5, 10}

A = {2, **7, 8**, 3, 4, 9, 1, 6, 5, 10}

….. giải thuật này khá cồng kềnh nên lượt bớt thầy ơiii

A = {1, 2, 3, 4, 5, 6, 7, **8, 9,** 10}

A = {1, 2, 3, 4, 5, 6, 7, 8, **9, 10**}

* Trong Bubble sort, ta so sánh từng cặp số và lặp lại vài vòng cho đến khi tất cả các cặp số được sắp xếp.
* Hàm:

void bubbleSort (int a[], int size) {

int temp, k, pass;

for (pass = 1; pass <= size - 1; pass++) {

for (k = 0; k < size - pass ; k++) {

if (a[k] > a[k+1]) {

temp = a[k];

a[k] = a[k+1];

a[k+1] = temp;

}

}

}

return;

}

**Problem 4.** Estimate the computational complexity of above algorithms.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Best | Average | Worst |
| [Selection Sort](http://geeksquiz.com/selection-sort/) | Ω(n^2) | θ(n^2) | O(n^2) |
| [Bubble Sort](http://geeksquiz.com/bubble-sort/) | Ω(n) | θ(n^2) | O(n^2) |
| [Insertion Sort](http://geeksquiz.com/insertion-sort/) | Ω(n) | θ(n^2) | O(n^2) |