

Thuật toán sắp xếp nổi bọt(bubble sort) : so sánh 2 phần tử liền kề và đưa phần tử lớn nhất nổi về cuối dãy.

Độ phức tạp : $O(n^2)$.

Code:

```
Void bubblesort( int a[], int n)
{
    for ( int i=0; i<n; i++){
        for( int j=i+1; j<n; j++){
            if(a[i]>a[j]){
                swap(a[i], a[j]);
            }
        }
    }
}
```

Ưu điểm : đơn giản.

Nhược điểm : chậm, ko hiệu quả với mảng lớn.

Độ phức tạp ngang selection sort, nhưng vẫn chậm hơn insertion sort.

Tìm kiếm nhị phân (Binary Search) là một thuật toán tìm kiếm trên mảng đã được sắp xếp.

Ý tưởng của thuật toán:

Chia đôi mảng:

Nếu phần tử cần tìm nằm ở giữa, trả về vị trí của nó.

Nếu phần tử cần tìm nhỏ hơn phần tử giữa, tìm trong nửa trái.

Nếu phần tử cần tìm lớn hơn phần tử giữa, tìm trong nửa phải.

Gọi đệ quy với nửa mảng phù hợp.

Nếu không tìm thấy, trả về -1.

Code:

```
int binarySearch(int arr[], int left, int right, int x) {  
    if (left > right)  
        return -1; // Không tìm thấy  
  
    int mid = left + (right - left) / 2; // Tránh tràn số  
  
    if (arr[mid] == x)  
        return mid; // Tìm thấy phần tử  
  
    if (arr[mid] > x)  
        return binarySearch(arr, left, mid - 1, x); // Tìm bên trái
```

```
    return binarySearch(arr, mid + 1, right, x); // Tìm bên phải  
}
```

Ưu điểm: code gọn, dễ hiểu, độ phức tạp $O(\log(n))$.

Nhược điểm : tốn bộ nhớ, chỉ dung cho mảng đã sắp xếp.