Full Name: Nguyễn Anh Tuấn - 17000909

Class: K62A3

1. Thuật toán sắp xếp chọn

* Viết mã giả cho thuật toán:

b = a;

b.sort();

for i=0 to n-1 do:

a[a.index(b([i])] = a[i];

a[i] = b[i];

* Sau mỗi lần duyệt phần tử, thuật toán tìm ra được phần tử nhỏ nhất trong mảng rồi sắp xếp lại lần lượt, như vậy đến phân tử cuối cùng, hiển nhiên nó sẽ là phần tử lớn nhất trong mảng bởi n-1 phần tử ban đầu đã được sắp xếp và so sánh với phần tử cuối

Như vậy, không cần phải duyệt qua phần tử cuối của mảng mà chỉ cần duyệt n-1 phần tử đầu tiên.

* Trường hợp tốt nhất, thời gian chạy là O1, trường hợp xấu nhấy thời gian chạy là On, n càng lớn thì thời gian chạy càng lớn

1. Bài toán tìm kiếm

* Ý tưởng: duyệt lần lượt các phần tử trong mảng và so sánh với v. Nếu a[i] = v. Trả về i
* Mã giả:

For i = 0 to n-1 do:

if a[i] == v:

print(i);

break;

* Bất biến của vòng lặp: a[i] = v, đúng nếu a có chứa v, i có, nếu a không có v thì i = 0
* Thời gian thực hiện tốt nhất là O1, dài nhất là On, n càng lớn thời gian càng lớn

3.

105.

f(n)=n+log⁡n,g(n)= √n

Ta thấy

√n≤n ≤n+log⁡n

Chọn c = 1;n\_0=10 ⟹g(n)=O(f(n))

106.

f(n)=2〖(log⁡n)〗^2,g(n)=log⁡n+1

Ta có log⁡n+1 ≤2 log⁡n ≤2〖(log⁡〖n)〗〗^2

Vậy khi chọn c=1,n\_0=10

⟹g(n)=O(f(n))