Bài 1: Sắp xếp theo BigO

2 mũ 10

3n+100logn

3n+100logn = nlogn

4n

2 mũ logn

n mũ 2 + 10n

n mũ 3

2 mũ n

Bài 2:

Cách 1:

Function power (n)

Int power = 1;

For chạy từ 1 đến n

power \*= 2

Return power

Cách này độ phức tạp là O(n)

Cách 2:

Function power (res, n)

if(n == 0) return 1

if(n is odd) return res\*power(res\*res, n/2)

if(n is even) return power(res\*res, n/2)

Cách này độ phức tạp là O(logn)

Bài 3:

Int arr[MAX]

Int head = 0, rear = -1

Function add(value)

if(rear < MAX)

rear ++

arr[rear] = value

Function remove

if(head <= rear)

Head++

Function search(value)

for(i: head to rear)

if(value == arr[i]) return 1

return 0

Function print

for(i: head to rear)

print arr[i]

return 0

Bài 4: Class linkedList đã được định nghĩa sẵn có head và rear

linkedList queue

Function add(value)

node tmp = new node(value)

if(queue != empty)

queue.rear.next = tmp

else

queue.head = tmp

Function remove

if(queue != empty)

queue.head = queue.head.next

Function search(value)

while(tmp != queue.rear)

print tmp

if(value == tmp.value) return 1

return 0

Function print

while(tmp != queue.rear)

print tmp

tmp = tmp.next

Bài 5:

Int arr[MAX]

Int head = -1

Function add(value)

if(rear < MAX)

head++

arr[head] = value

Function remove

if(head >= 0)

head–

Function search(value)

for(i: 0 to head)

if(value == arr[i]) return 1

return 0

Function print

for(i: 0 to head)

print arr[i]

return 0

Bài 6: Class linkedList đã được định nghĩa sẵn có head và rear

linkedList stack

Function add(value)

node tmp = new node(value)

if(stack != empty)

tmp.next = queue.head.next

queue.head = tmp

Function remove

node tmp = stack.head

while(tmp && tmp.next != stack.rear) tmp = tmp.next

stack.rear = tmp

tmp.next = null

Function search(value)

while(tmp != stack.rear)

print tmp

if(value == tmp.value) return 1

return 0

Function print

while(tmp != stack.rear)

print tmp

tmp = tmp.next