Nguyễn Lê Đức Nhật

24120403

Bài 1: Fibonacci

Base case: •Với n = 0 hoặc n = 1 thì return n. Vì fibo[0] = 0,fibo[1] = 1;

Recursive case:

Nếu n > 1 thì gọi đệ quy fibo[n – 1] + fibo[n – 2]

// (fibo[n] = fibo[n – 1] + fibo[n – 2])

Bài 2: Factorial

•**Base case:** Nếu (n = 0 || n = 1) thì return 1 vì 0! = 1! = 1

•**Recursive case:**

Nếu n > 1 thì gọi hàm đệ quy n * factorial(n – 1)

// vì n! = n * (n – 1)!

Bài 3: Generate Binary String

•Base case:

+ Nếu str.size() == n (tức chuỗi str đã có chiều dài n) thì ta in ra kết quả và return để hàm backtrack đến bước trước đó.

+ Char c[2] = {‘0’,’1’} // vì đây là xâu nhị phân nên mỗi kí tự chỉ có thể 2 giá trị là 0 và 1.

Lặp 1 vòng for từ 0 đến 1,thêm kí tự c[i] vào xâu str sau đó gọi đệ quy (n,str)

Khi str.size() == n , tức là gặp base case thì hàm sẽ in ra xâu str có độ dài n sau đó backtrack để tiếp tục xử lí,**str.pop_back()** chính là bước backtrack

Bài 4: Tower of Hanoi

Bài 5: is Sorted array

•**Base case:** Nếu(n = 1 || n = 0) return true vì mảng có 1 hoặc 0 phần tử thì luôn được sắp xếp

// check:nếu phần tử đầu tiên của mảng lớn hơn phần từ thứ hai thì return false

•**Recursive case:** gọi đệ quy (arr + 1,n – 1),lúc này tăng chỉ số của mảng lên 1 đơn vị và giảm số phần tử của mảng đi 1 đơn vị.

Bài 6: N-Queens