

# NHẬP MÔN PHÂN TÍCH THUẬT TOÁN

(120 phút, không dùng tài liệu)

## Câu 1

Thuật toán tìm kiếm sau đây xác định xem có mẫu tin nào trong mảng  $a$  (gồm  $n$  mẫu tin  $a[1], a[2], \dots, a[n]$ ) có khóa là  $K$  hay không. Giả sử các mẫu tin được chọn ngẫu nhiên có khóa đôi một phân biệt.

```
I := 1;
success := 0;
while (i<=n) and (success=0) do
    if K=a[i].Key then
        success := i;
    endif;
    i := i+1;
endw;
```

Hãy ước lượng số phép toán (phép gán và phép so sánh số học, phép so sánh khóa) trung bình của thuật toán trong các trường hợp sau:

- a) Giả sử  $a[i].Key = i^2 + i$  và  $K$  được chọn ngẫu nhiên trong các số nguyên từ  $-n^3$  đến  $n^3 + n$ . Trường hợp này khả năng “tìm có” (tức  $success \neq 0$ ) có cao hơn khả năng “tìm không có” hay không?
- b) Giả sử  $a[i].Key = i^3$  và  $K$  là giá trị ghi trên một lá phiếu nào đó được chọn ngẫu nhiên từ các lá phiếu được ghi các số như sau:
- $n+1$  phiếu có giá trị  $-2^n, -2^{n-1}, -2^{n-2}, \dots, -2, -1$
  - gồm có  $4^n$  phiếu có giá trị là 1
  - gồm có  $4^{n-1}$  phiếu có giá trị là 2
  - gồm có  $4^{n-2}$  phiếu có giá trị là 3
  - ...
  - gồm có  $4^1$  phiếu có giá trị là  $n$

(Tất cả những phiếu này được xáo trộn để việc bốc phiếu mang tính ngẫu nhiên)

## Câu 2

Giả sử  $n$  là một số nguyên dương. Xem thuật toán trong đoạn chương trình C sau đây:

```
float Alpha(float x, long n) {
    long i=1; float z=0;
    while(i<=n){
        long j=1; float t=1;
        while(j<=i){
            t = t*x;
            j = 2*j;
        }
        z = z + i*t;
        i = i + 1;
    }
    return z;
}
```

Hãy xác định theo  $n$  số phép gán và số phép so sánh được thực hiện trong thuật toán. Ước lượng độ phức tạp của thuật toán.