Phân tích và Thiết kế thuật toán
Bài tập tuần 4

Nhóm 11
Lê Hoài Thương - 21520474
Nguyễn Hoàng Hải - 21522034

Mục lục

0.1 Bài 1

Question 1:

The range of a finite nonempty set of n real number S is defined as the difference between the largest and the smallest elements of S. For each representation of S given below, describe in English an algorithm to compute the range. Indicate the time efficiency classes of these algorithm using the most appropriate notation

Solution:

1. An unsorted array
   Để tìm Min và Max của mảng chưa sắp xếp, ta phải duyệt qua tất cả các phần tử.
   $\Theta(n)$
   $\Omega(n)$
   $O(n))$

2. A sorted array
   Khi mảng đã được sắp xếp, phần tử đầu và cuối của mảng là Min và Max.
   $\Theta(1)$
   $\Omega(1)$
   $O(1)$

3. A sorted singly linked list
   Đối với danh sách liên kết đơn đã sắp xếp, phần tử head và tail là Max và Min. Tuy nhiên, để truy cập được phần tử tail thì phải duyệt qua hết linked list.
   $\Theta(n)$
   $\Omega(n)$
   $O(n)$

4. A binary search tree
   Đối với cây nhị phân cân bằng, thao tác lấy nút lát trái nhất (phần tử nhỏ nhất) và nút lá phải nhất (phần tử lớn nhất) sẽ mất log(n) bước. Còn đối với cây nhị phân không cân bằng thì trường hợp xấu nhất sẽ phải duyệt toàn bộ cây.
   $\Theta(\log(n))$
   $\Omega(\log(n))$
   $O(n)$

0.2 Bài 2

Question 2

Lighter or heavier? You have $n > 2$ identical looking coins and a two-pan balance scale with no weights. One of the coins is a fake, but you do not know whether it is lighter or heavier than the genuine coins, which all weigh the same. Design a $\Theta(1)$ algorithm to determine whether the fake coin is lighter or heavier than the others.

Solution: Chia n đồng xu thành 3 phần có số lượng bằng nhau. Số xu còn dư thì bỏ qua 1 bên. Sau 2 lần cân lần lượt các chồng xu với nhau, có 2 trường hợp:

- Trường hợp 1: 3 chồng xu nặng bằng nhau
  Khi đó 1 trong những đồng xu dư ra là giả. Lấy thêm 1 đồng xu từ 3 chồng xu rồi tiến hành đo. Sau nhiều nhất 2 lần cân nữa thì xác định được đồng xu giả nặng hơn hay nhẹ hơn.

- Trường hợp 2: 3 chồng xu nặng không bằng nhau:
  Có 1 chồng xu có khối lượng khác 2 chồng xu còn lại. Nếu chồng xu đó nhẹ hơn thì đồng xu giả nhẹ hơn đồng xu thật và ngược lại.

Ta thấy chỉ cần nhiều nhất là 4 lần cân để xác định được đồng xu giả nặng hơn hay nhẹ hơn đồng xu thật không phụ thuộc vào n. => Thuật toán trên có độ phức tạp $O(1)$;

0.3 Bài 3

Question 3

Consider the following version of an important algorithm that we will study later in the book.

Algorithm 1: GE(A[0..$n-1$, 0..n])

//An n × (n + 1) matrix A[0..$n-1$, 0..n] of real numbers
for $i \leftarrow 0$ ***to*** $n-2$ **do**
    for $j \leftarrow i+1$ ***to*** $n-1$ **do**
        for $k \leftarrow i$ ***to*** n **do**
            $A[j,k] \leftarrow A[j,k] - A[i,k] * A[j,i]/A[i,i]$

a Find the time efficiency class of this algorithm.

b What glaring inefficiency does this pseudocode contain and how can it be eliminated to speed the algorithm up?

Solution:

Answer a: Cả 3 vòng for đều duyệt qua tối đa $n+1$ giá trị, do chúng lồng vào nhau nên độ phức tạp là $O(n^3)$.

Answer b: Xét thấy, sau khi thực hiện toàn bộ thuật toán, các phần tử **A[i,j]** mà $(i > j)$ của ma trận **A** sẽ có giá trị bằng 0, các phần tử còn lại của ma trận **A** không có gì thay đổi. Như vậy, có thể thấy rằng input của thuật toán là một ma trận **A** và output của thuật toán là ma trận **A** đã biến đổi sao cho **A[i, j] = 0** với $(i > j)$ và các phần tử còn lại được giữ nguyên giá trị ban đầu. Từ đó, thay vì phải sử dụng 3 vòng lặp for cùng với công thức phức tạp, ta có thể sử dụng thuật toán đơn giản hơn là gán các giá trị **A[i, j]** bằng 0 với $(i > j)$. Theo cách làm này, thuật toán chỉ cần duyệt từng phần tử **A[i, j]** $(i > j)$ của ma trận **A** và gán chúng bằng giá trị 0, độ phức tạp lúc này là $O(n^2)$.

Algorithm 2: GE(A[0..$n-1$, 0..n])

//An n × (n + 1) matrix A[0..$n-1$, 0..n] of real numbers
for $i \leftarrow 0$ ***to*** $n-1$ **do**
    for $j \leftarrow 0$ ***to*** $i-1$ **do**
        $A[i,j] = 0$