***CS112.M11.KHTN - Nhóm 8 - Bài tập Độ phức tạp thuật toán***

**Câu 1: Phân tích độ phức tạp thời gian thuật toán quickselect.**

Quickselect là thuật toán tìm phần tử nhỏ nhất thứ k của một mảng chưa sắp xếp. Dựa trên ý tưởng của quicksort, chỉ khác ở điểm:

if k-1 = pivotIndex

return list[k-1]

else if k-1 < pivotIndex

right := pivotIndex - 1

else

left := pivotIndex + 1

Do đó,

T(n) = T(pivotIndex) + O(n)

Hay T(n) = T(n - pivotIndex) + O(n)

Vì pivotIndex có kỳ vọng bằng n/2, ta có

T(n) = T(n/2) + O(n)

Áp dụng định lý master:

a = 1, b = 2, c = 1

=> 1 = c = loga(b) = log2(2)

=> T(n) = O(n)

Trong trường hợp tốt nhất, pivotIndex đầu tiên tìm thấy = k - 1, độ phức tạp O(n).

Trong trường hợp tệ nhất, tất cả các phần tử bằng nhau, độ phức tạp O(n^2).

**Câu 2: Dùng phương pháp Potential tính độ phức tạp của hàm push\_back.**

****

Φ(v) = 2\*v.size() - v.capacity()

với Φ(v) là potential function của các bộ trong vector

Vì chiến lược thay đổi capacity luôn làm cho size của vector ít nhất bằng một nửa, nên hàm này luôn không âm. Và ta có giá trị khởi tạo của size và capacity đều bằng 0 nên hàm potential ở trạng thái ban đầu bằng 0.

Gọi Ci là chi phí (thời gian) trung bình cho lần push\_back thứ i.

TH1: không làm tăng capacity

Ci = 1 + Φi - Φ(i-1)

= 1 + {2\*[size+1] - capacity} - {2\*size - capacity}

= 1 + 2

= 3

TH2: làm tăng capacity

capacity = 2\*capacity(i-1), size(i-1) = capacity(i-1), sizei = capacity(i-1) + 1

Ci = capacity(i-1) + 1 + Φi - Φ(i-1)

= capacity(i-1) + 1

+ {2\*[capacity(i-1) + 1] - 2\*capacity(i-1)}

- {2\*capacity(i-1) - capacity(i-1)}

= 3

Vậy hao phí thời gian trung bình của hàm push\_back vector là hằng số ⇔ O(1) :)

**Câu 3: Theo thuật toán tìm cách đặt 8 quân hậu trên đề, tìm kỳ vọng số lần khởi tạo bàn cờ.**

Gọi xsMotLan là xác suất hoàn thành bài toán trong một lần khởi tạo.

slkt là số lần khởi tạo bàn cờ để hoàn thành bài toán.

Do mỗi lần khởi tạo, thuật toán đều hoạt động giống nhau, không sử dụng thông tin từ vòng lặp trước nên

P(slkt = 1) = xsMotLan = P(slkt = 2| slkt > 1) = P(slkt = 3| slkt > 2)

= … = P(slkt = n| slkt > n-1)

Bí. Hết. Nhờ Minh giải hộ. :(