**Group 6**

**Bài tập phân tích độ phức tạp thuật toán**

**Câu 1:** Phân tích độ phức tạp của thuật toán quickselect

[Quickselect](https://en.wikipedia.org/wiki/Quickselect) là một thuật toán tìm phần tử nhỏ thứ k trong một danh sách không có thứ tự. Nó liên quan đến thuật toán quicksort.

Có điểm khác đó là:

* if pivot\_index > k-1:

return quickselect(start, pivot\_index – 1)

* elif pivot\_index < k-1:

return quickselect(pivot\_index + 1, end)

* else:

return list[k – 1]

Độ phức tạp thời gian: T(n) = T(n/2) + O(n)

Áp dụng định lý Master, ta có: a = 1, b = 2, c = 1.

Ta lại có: logb(a) = log2(1) = 0 < c = 1

Do đó T(n) = O(n).

Vậy độ phức tạp của thuật toán quickselect là O(n).

**Câu 2:** Ngoài phương pháp aggregate còn có phương pháp potiental. Hãy dùng phương pháp đó để tính độ phức tạp của hàm thêm vào trong cấu trúc dữ liệu vector.

**Ta có hàm potential: Φ = 2n – N**

Áp dụng vào bài này, với:

n: số lượng phần tử (size)

N: kích thước vector (capacity). Kích thước là số lượng phần tử nhiều nhất mà vector đó từng chứa còn số lượng phần tử là số lượng phần tử mà vector đó hiện tại đang chứa.

Ta có hàm potential trong bài toán này như sau: Φ(v) = 2\*v.size() – v.capacity()

Ta có khi n=N thì N sẽ được double lên. Do đó ta luôn có n ≥ ½ \* N (size ≥ ½ \* capacity) nên hàm potential luôn không âm.

Khởi tạo hàm potential bằng 0

Gọi Ti là thời gian cho lần push\_back thứ i.

***Trường hợp 1****: Push\_back nhưng không làm tăng capacity*

Ti = 1 + Φ i – Φ i-1 = 1 + (2\*(size+1) – capacity) – (2\*size - capacity) = 1 + 2 = 3

***Trường hợp 2:*** *Push\_back và làm tăng capacity.*

T­i = capacityi-1 + 1 + Φ i – Φ i-1

= capacityi-1 + 1 + (2\*(capacityi-1 + 1) - 2\*capacityi-1) - (2\*capacityi-1 – capacityi-1)

= 3

Từ 2 trường hợp trên ta kết luận độ phức tạp của hàm push\_back là O(1) (constant).

Link nhóm tham khảo: https://people.engr.tamu.edu/andreas-klappenecker/csce411-s14/csce411-amortized2.pdf

**Câu 3:** Các bạn hẳn còn nhớ bài toán 8 hậu, rằng tìm một cách đặt 8 quân hậu sao cho không con nào ăn được nhau. Nếu thuật toán của của ta có các bước như sau:

Bước 1: Khởi tạo bàn cờ.

Bước 2: Nếu đã đặt đủ quân hậu, xuất đáp án.

Bước 3: Tìm tất cả vị trí trong hàng có để đặt được quân hậu sao cho không quân hậu nào ăn được quân hậu khác.

Bước 4: Nếu không có vị trí nào thỏa mãn, đi đến bước 1.

Bước 5: Nếu có, chọn ngẫu nhiên một vị trí, di chuyển đến hàng tiếp theo và quay lại bước 3.

Thì kỳ vọng bạn sẽ phải khởi tạo bao nhiêu bàn cờ

**Lời giải:**

Độ phức tạp về thời gian của bài toán này là T(n) = O(n!)

Người ta đã tìm ra tất cả là 92 cách giải quyết bài toán này trên bàn cờ 8x8. Tuy nhiên, nếu chỉ xét các trường hợp tổng quát (hay nói cách khác, xem 2 trường hợp có thể cùng nhận được bằng cách xoay bàn cờ hoặc dùng phép đối xứng là như nhau), thì trên bàn cờ có tất cả là 12 cách giải. Trong đó, chỉ có cách giải đối xứng đã biểu diễn ở trên là có 3 cách giải tương tự, các trường hợp tổng quát khác đều có thể tạo ra thêm 7 cách giải tương đương, vì vậy tổng số cách sắp xếp là 11x8 + 1x4 = 92 .

Do đó số bàn cờ phải khởi tạo ít nhất là: 8! / 92 = 438,261

Link tham khảo về độ phức tạp của bài toán 8 quân hậu:

https://iq.opengenus.org/8-queens-problem-backtracking/#:~:text=For%20thr%20given%20problem%2C%20we,approach%20is%20O(N!).