BÁO CÁO BÀI TẬP LỚN
Môn: Project1

Mục lục

Lời mở đầu

Vì bucket sort sử dụng thuật toán phụ là insertion sort, ta có thể chứng minh thời gian của bucket sort là

$$T(n) = \Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)$$

với n_i là số lượng phần tử trong thùng thứ i.

Như vậy thời gian trung bình của thuật toán này sẽ là

$$\begin{aligned} \mathrm{E}[T(n)] &= \mathrm{E}\left[\Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)\right] \\ &= \Theta(n) + \sum_{i=0}^{n-1} \mathrm{E}\left[O(n_i^2)\right] \\ &= \Theta(n) + \sum_{i=0}^{n-1} O\left(\mathrm{E}\left[n_i^2\right]\right) \end{aligned}$$

Vì số lượng phần tử trong mỗi thùng phụ thuộc vào số thùng được cung cấp, và do số phần tử trong thuật toán insertion sort có liên quan tới số thùng, nên thuật toán này có những thời trường hợp tốt và xấu khác nhau, thay đổi từ O(1) tới $O(n^2)$, chủ yếu do đầu vào dữ liệu có phù hợp với insertion sort hay không.

Để làm rõ các tường hợp của thuật toán Bucket sort, chúng em sẽ chạy thuật toán với những đầu vào khác nhau, qua đó rút ra kết luận những trường hợp nào nên sử dụng Bucket sort. Trong phần demo sau, chúng em sử dụng các số điện thoại ở các tỉnh thành trên cả nước. Nguồn dữ liệu lấy từ trang thongtinlienlac.com.

Chương I - Thời gian của Bucket Sort

1.1. Dữ liệu đầu vào là ngẫu nhiên

Trong phần này, chúng em sử dụng 100 000 số điện thoại (10^6 số) được lấy ngẫu nhiên từ các tỉnh thành trên cả nước.

Vì các đầu số điện thoại này là khác nhau ở mỗi tỉnh thành có dạng 02x gồm 11 chữ số, nên có thể coi các số này là trong khoảng từ $2*10^9$ đến $(10*10^9-1)$. Để đơn giản hóa, ta bỏ 2 chữ số "02" ở đầu đi, nên dữ liệu sẽ có giá trị từ 0 đến $(10^9 - 1)$

Chúng em chạy thuật toán với các số lượng thùng là từ 10^3 đến 10^7 thùng. Chạy với 100 bộ dữ liệu khác nhau (cùng lấy ngẫu nhiên từ kho 10^7 phần tử) rồi lấy trung bình thời gian chạy.

Kết quả:

Size	Time (ms)

Như vậy ta có thể thấy, thời gian sắp xếp của Bucket sort, ngoài việc phụ thuộc vào dữ liệu đầu vào (do sử dụng insertion sort) còn phụ thuộc vào số lượng *thùng chứa* sử dụng để sắp xếp.

1.2. Đầu vào của Bucket Sort đã được lọc

Trong phần này vẫn sử dụng 100 000 số điện thoại (10^6 số), nhưng được lấy từ 1 tỉnh cụ thể, ví dụ nếu là Hà Nội thì số điện thoại có dạng 024 xxxx xxxx, sau khi convert sẽ có dạng 4 xxxx xxxx.

Vẫn thực hiện thuật toán với số lượng thùng như trên, ta có bảng sau:

Size	Time (ms)

Trường hợp này cho thấy tốc độ có thay đổi khá rõ ràng, nguyên nhân là do chữ số 4 ở đầu đã cố định phần lớn các phần tử vào các thùng vị trí 4x. Do đó thời gian sắp xếp các số này (mà thực chất là sắp xếp bằng insertion sort) tăng lên đáng kể.

Chương 2. Lưu ý khi dùng Bucket Sort

Như phần trên đã trình bày, bucket sort là một thuật toán sắp xếp khá nhanh với thời gian O($n+k$), tức là thời gian tuyến tính. Tuy nhiên hiệu quả của thuật toán này phụ thuốc vào nhiều yếu tố như số lượng thùng cấp phát, và miền giá trị của phần tử đầu vào, qua đó cần xác định được số lượng phần tử đầu vào và ước lượng khoảng giá trị của chúng.