

REPORT_24120184_WEEK2

Exercise 1:

Yêu cầu đề bài:

- Cho 1 mảng gồm n phần tử (kiểu dữ liệu int) và target k, viết chương trình tìm phần tử đầu tiên có giá trị là k, sử dụng hàm Linear Search . Nếu k tìm được trả về index, không tìm được trả về -1.

Hướng giải quyết:

- Viết hàm Linear Search để tìm theo yêu cầu của đề bài.
- Hàm:
 - Sử dụng vòng lặp duyệt qua giá trị của từng phần tử , khi tìm được phần tử có giá trị bằng k trả về index, khi duyệt hết phần tử của mảng không tìm được trả về -1.

Độ phức tạp:

- Do sử dụng 1 vòng lặp trong hàm nên độ phức tạp của hàm là $O(n)$.

Exercise 2:

Yêu cầu đề bài:

- Cho 1 mảng gồm n phần tử (kiểu dữ liệu int) và target k, viết chương trình tìm phần tử đầu tiên có giá trị là k, sử dụng hàm sentinelLinear Search . Nếu k tìm được trả về index, không tìm được trả về -1.

Hướng giải quyết:

Viết hàm sentinelLinearSearch để tìm theo yêu cầu đề bài.

Hàm:

Ý tưởng:

- Thêm k vào phần tử cuối cùng của mảng, để giảm độ phức tạp, loại bỏ 1 phép so sánh sau mỗi vòng lặp.
 - Đầu tiên kiểm tra phần tử cuối cùng có bằng k hay không => Có thì trả về index.
 - Lưu giá trị của phần tử cuối , gán phần tử cuối bằng giá trị của k.
 - Duyệt i chạy từ 0 -> n - 1, kiểm tra xem có phần tử nào bằng k hay không
 - Nếu có thì $i < n - 1$, không có thì $i = n - 1$. Trả lại giá trị của phần tử cuối
- ⇒ Trả về Index thỏa mãn đề bài.

Độ phức tạp:

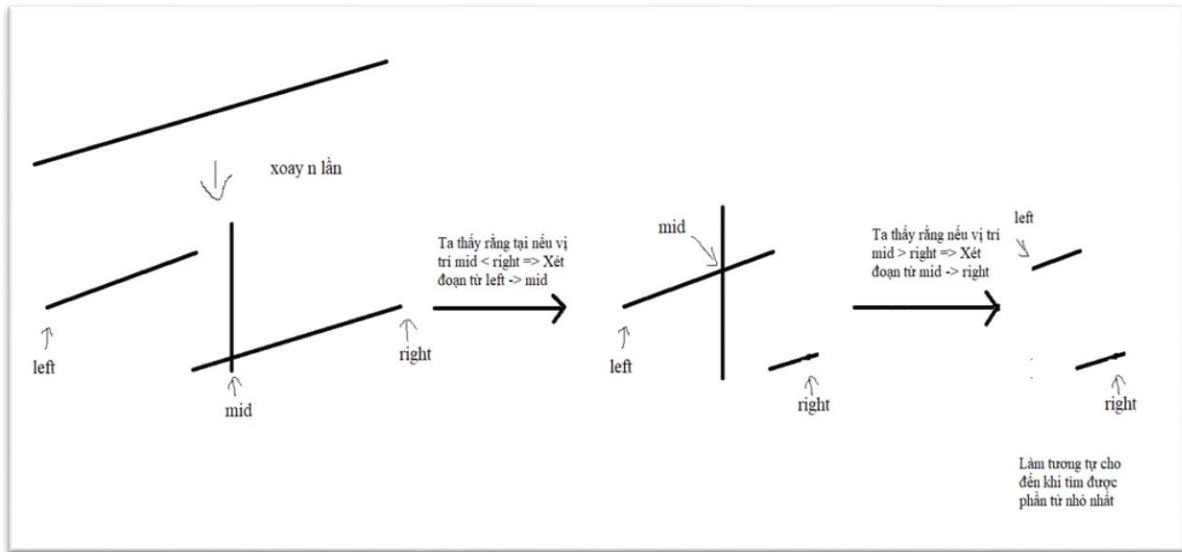
- Sử dụng 1 vòng lặp for nên sẽ có độ phức tạp $O(n)$, nhưng mỗi vòng lặp sẽ bớt 1 phép so sánh => tối ưu hơn linearsearch.

Exercise 3:

Yêu cầu đề bài:

- Cho 1 mảng gồm n phần tử đã sắp xếp theo thứ tự tăng dần và đã xoay trong khoảng từ 1 \rightarrow n lần bất kì.
- Tìm phần tử nhỏ nhất trong mảng đảm bảo thuật toán có độ phức tạp $O(\log n)$.

Hướng giải quyết:



➤ Ý tưởng:

- Cho 2 vị trí nằm ở 2 biên left và right, vị trí $\text{mid} = (\text{left} + \text{right})/2$, So sánh vị trí mid với right, nếu $\text{mid} < \text{right} \rightarrow$ Xét tìm min ở đoạn từ left - mid, ngược lại xét ở đoạn từ mid - right.
- Hoặc có thể so sánh mid với left, $\text{mid} < \text{left}$ tìm min ở đoạn từ left - min, ngược lại xét ở đoạn từ mid - right.

➤ Hàm tìm min:

+Cho 2 biến trỏ đến 2 vị trí left và right, và $\text{mid} = (\text{left} + \text{right})/2$;

+Sử dụng vòng lặp với điều kiện $\text{left} < \text{right} - 1 \Rightarrow$ Tức là dừng khi xét chỉ còn 2 phần tử.

+Nếu $\text{min} < \text{right}$: gán $\text{right} = \text{mid}$.

+Ngược lại: gán $\text{left} = \text{mid}$.

+Sau khi kết thúc vòng lặp chỉ còn 2 phần tử, thực hiện phép so sánh và trả về kết quả nhỏ nhất.

Độ phức tạp:

- Do sau mỗi vòng lặp đều cập nhật vị trí left hoặc right bằng với mid tương ứng vòng lặp sau chỉ làm việc với nửa mảng còn lại, tức là $n/2 \Rightarrow$ Độ phức tạp của thuật toán $O(\log n)$.

Exercise 4:

Yêu cầu đề bài:

- Một băng chuyền có các gói hàng được vận chuyển từ cảng này sang cảng khác trong một vài ngày.
- Gói hàng thứ i có trọng lượng là $weights[i]$ (coi $weights$ là 1 mảng gồm lần lượt khối lượng của các kiện hàng). Mỗi ngày chất lên băng chuyền khối lượng hàng nhất định, không được chất tổng khối lượng của kiện hàng lớn hơn khối lượng tối đa tàu cho phép.
- Tìm khối lượng tối thiểu của tàu chứa khiến tất cả kiện hàng được vận chuyển đúng với số ngày đề ra.

Hướng giải quyết:

- Vì mỗi ngày sẽ vận chuyển được ít nhất 1 kiện hàng, nên khối lượng tối thiểu của tàu chở phải lớn hơn kiện hàng có khối lượng lớn nhất.
- Lần lượt tăng khối lượng tàu có thể chở được, xem xét lần lượt liệu đã chở hết tất cả các kiện hàng trong số ngày nhất định hay chưa và trả về khối lượng tối thiểu mà tàu có thể chở.
- Viết hàm:
+1 vòng lặp for duyệt tìm kiếm kiện hàng có khối lượng lớn nhất và gán cho t (khối lượng tối thiểu mà tàu chở được).

+Bắt đầu vòng lặp: t tăng dần dần.

- Duyệt qua lần lượt các kiện hàng, biến temp dùng biểu thị khối lượng tổng 1 ngày có thể vận chuyển được
- Nếu khối lượng 1 ngày $> t \Rightarrow$ reset khối lượng tổng = 0, số ngày vận chuyển tăng lên 1, bỏ kiện hàng vừa thêm cho ngày tiếp theo vận chuyển

+Hết vòng lặp kiểm tra lại số ngày vận chuyển đã thỏa hay không, vận chuyển vừa đủ \Rightarrow trả về t thỏa mãn

+Nếu ngày dùng để xét tạm vẫn nhỏ hơn số ngày đề ra, thì không tìm được t , do số ngày vận chuyển lại lớn hơn số lượng kiện hàng trả về -1.

Độ phức tạp:

- Khi tìm khối lượng kiện hàng lớn nhất $\Rightarrow O(n)$.
- Vòng lặp while, trường hợp xấu nhất có thể xảy ra là khi t từ khối lượng lớn nhất đến tổng khối lượng các kiện hàng.
- Với mỗi giá trị t thì lặp lại n lần
 \Rightarrow Độ phức tạp: $O(n \times (S - m))$ với S là tổng khối lượng các kiện hàng, m là khối lượng của kiện hàng có khối lượng lớn nhất.

Exercise 5:

Yêu cầu đề bài:

- Cho 1 mảng gồm có số nguyên dương và 1 số nguyên dương target, trả về chiều dài tối thiểu của mảng con trong đó nó có tổng lớn hơn hoặc bằng target. Nếu không tìm được mảng con trả về 0.

Hướng giải quyết:

- Sử dụng 2 con trỏ và nó sẽ di chuyển phụ thuộc vào tổng mảng con ra sao. Mảng con được đánh dấu ở đầu bởi con trỏ 1 và cuối bởi con trỏ 2.
- Khi 2 con trỏ đều trỏ ở vị trí đầu, thì mảng con gồm 1 phần tử duy nhất đó là giá trị của phần tử đầu. Ta so sánh tổng mảng con với target:
- Nếu lớn hơn thì ta cần lưu độ dài mảng con và giảm chiều dài của mảng con nhằm tìm kiếm tiếp tục xem liệu có mảng con nào có độ dài nhỏ hơn vẫn thỏa mãn điều kiện đề bài
- Nếu nhỏ hơn thì ta tăng độ dài mảng con bằng cách tăng chiều dài của mảng con.
- Viết hàm:

-Đặt 2 con trỏ trỏ đến vị trí đầu, giá trị sum bằng phần tử ban đầu.

-1 biến check để kiểm tra có mảng con nào thỏa điều kiện đề bài.

-Vòng lặp với điều kiện dừng ($left > right$ hoặc $right < n$):

+Nếu sum của mảng con lớn hơn hoặc bằng target thì đánh dấu đã tìm được mảng con thỏa, so sánh với mảng con trước đó liệu có kích thước nhỏ hơn hay không, và gán giá trị min bằng kích thước mảng con nhỏ hơn, giảm độ dài mảng con để tìm kiếm mảng con nhỏ hơn ($left - 1$).

+Nếu sum nhỏ hơn target, tăng kích thước mảng con nhằm tìm mảng con có tổng \geq target ($right + 1$).

-Sau vòng lặp kiểm tra biến check và trả về kích thước nhỏ nhất của mảng con có tổng \geq target

-Check = false trả về 0 do không tìm được mảng con phù hợp.

Độ phức tạp:

- Hàm chỉ sử dụng 1 vòng lặp nên sẽ có độ phức tạp là $O(n)$.

Exercise 6:

Yêu cầu đề bài:

- Cho 1 mảng số nguyên đã sắp xếp và target tổng, tìm nếu chỉ tồn tại 2 số trong mảng liệu có tổng bằng target hay không?

Hướng giải quyết:

- Ta chọn 2 con trỏ, 1 con trỏ ở đầu mảng, 1 con trỏ ở cuối mảng do tính chất của mảng tăng dần, ta có thể tìm theo cách sau:

+Nếu tổng 2 con trỏ lớn hơn target, con trỏ phải lùi về do mảng tăng nên nghĩa là khi con trỏ phải lùi về thì tổng sẽ tiến gần đến target

+Ngược lại nếu tổng 2 con trỏ nhỏ hơn target, con trỏ trái tiến lên => tổng sẽ tiến gần đến target.

- Hàm:

-Cho vị trí 2 con trỏ ở đầu và cuối mảng.(left và right)

-Chạy vòng lặp với điều kiện $left < right$:

+Nếu tổng 2 con trỏ bằng target trả về true.

+Nếu tổng 2 con trỏ lớn hơn target: $right - 1$.

+Nếu tổng 2 con trỏ nhỏ hơn target: $left + 1$.

-Sau khi lặp nếu không tìm được cặp thỏa mãn trả về false.

Độ phức tạp:

- Vì chỉ chạy 1 dòng for nên độ phức tạp là $O(n)$.

Exercise 7:

Yêu cầu đề bài:

- Cho 1 mảng các số nguyên, tìm trả về bộ ba sao cho tổng bộ ba đó bằng 0. Kết quả xuất không tồn tại cặp trùng.

Hướng giải quyết:

- Vì mảng ngẫu nhiên nên ta dùng hàm sort để sắp xếp lại mảng.
- Duyệt qua từng phần tử của mảng và cho đó là giá trị target
- Để tổng bộ ba bằng không ta quy bài toán về tổng 2 phần tử bất kì bằng -target(giống như exercise 6).
- Sử dụng kỹ thuật 2 con trỏ để tìm 2 phần tử còn lại và xuất kết quả
- Viết hàm:

-Hàm sort nhằm sắp xếp lại mảng thành mảng tăng.

-Vòng lặp for đặt phần tử cần xét là target (ta cần so sánh với các phần tử trước đó nếu bằng thì bỏ qua để tránh sự trùng lặp).

-Khai báo vị trí 2 con trỏ: con trỏ 1 ở vị trí đầu + 1, con trỏ 2 ở vị trí cuối.

-Vòng lặp điều kiện (con trỏ 1 < con trỏ 2):

+Nếu tổng 2 con trỏ bằng - target thì xuất kết quả.

+So sánh tổng 2 con trỏ nếu lớn hơn -target thì con trỏ 2 dịch về bên trái tìm phần tử nhỏ hơn, ngược lại con trỏ 1 dịch về bên phải tìm phần tử lớn hơn.(tính chất mảng tăng).

Độ phức tạp:

- Hàm sort $O(n \log n)$.
- Hàm duyệt qua từng phần tử và vòng lặp while nên có độ phức tạp $O(n^2)$.

=>Độ phức tạp là $O(n^2)$.

Up folder lên Github:

