

Exercise 1

Dùng linearSearch kiểm tra từng phần tử trong mảng có bằng k không

Nếu có trả về vị trí đó. Nếu duyệt hết mảng vẫn không có k thì return -1.

Exercise 2

Mượn phần tử cuối mảng để đặt một lính canh (k) ở đó.

Duyệt mảng với chỉ một điều kiện cho tới khi gặp k. Trả lại phần tử cuối cùng

Nếu giá trị i nhỏ hơn n-1 hoặc phần tử cuối cùng =k thì return i.

Exercise 3

Khi xoay một mảng đã được sắp xếp thì khi chia đôi mảng đó ra ta luôn thu được ít nhất một mảng-> sử dụng binarySearch

Nếu $mid < right$ -> bên phải không có phần tử nào nhỏ hơn mid (mảng bên phải đã được sắp)-> $min = mid$ hoặc $min < mid$ -> gán $right = mid$

Nếu $mid \geq right$ -> mảng bên trái đã được sắp -> min chỉ có thể nằm bên mảng chưa được sắp-> $left = mid + 1$

Exercise 4

Tìm khoảng capacity với $min\ capacity = max\ weight$ (nếu 1 ngày đi một gói hàng),
 $max\ capacity = tổng\ tất\ cả\ weight\ của\ các\ gói\ hàng$ (nếu đi trong 1 ngày)

Đk: capacity nằm trong đoạn $[maxweight, sumweight]$, số ngày vận chuyển $\leq days$, mỗi ngày tổng weight $\leq capacity$ (*).??Tìm capacity nhỏ nhất thỏa đk(*)

Hàm đk canShip cho $currweight = 0$ và $currday = 1$ duyệt hết mảng weight, mỗi phần tử kiểm tra nếu $currweight + weight[i] >$ vượt quá tải trọng chuyển sang ngày mới ngược lại thì tang thêm $weight[i]$ vào ngày hiện tại. Trong quá trình duyệt nếu $currday > day$ thì return false (vượt quá số ngày quy định)

Sử dụng binarySearch để tìm min trong đoạn từ $[maxweight, sumweight]$

Nếu thỏa đk thì bên phải mid (lớn hơn mid) cũng thỏa đk -> $right = mid$

Nếu không thỏa đk thì bên trái mid (nhỏ hơn mid) cũng không thỏa $\rightarrow \text{left} = \text{mid} + 1$

Lặp lại đến khi $\text{left} = \text{right} = \text{mid}$ thì tìm được min thỏa đk (*)

Exercise 5

Duyệt tất cả mảng con. Tính tổng tất cả phần tử trong mảng nếu $< \text{target}$ thì không có mảng con nào có tổng bằng target, nếu tổng = target thì chỉ có duy nhất mảng n phần tử.

Tính riêng tổng các mảng con từ 1 phần tử đến n-1 phần tử, dùng biến để đếm độ dài mảng nếu có sumarr = target thì return độ dài mảng.

Exercise 6

Dùng 2 con trỏ đầu mảng và cuối mảng, tính tổng phần tử ở 2 con trỏ này nếu $> \text{target}$ thì giảm con trỏ bên phải một đơn vị ngược lại tăng con trỏ bên trái một đơn vị. Nếu có tổng = target thì return true ngược lại return false.

Exercise 7

Sử dụng 3 vòng lặp để duyệt tất cả bộ ba trong mảng. Tạo một mảng result để lưu giá trị các triplet có tổng bằng 0. Nếu có bộ ba có tổng bằng 0 thì kiểm tra xem nó đã tồn tại trong result chưa nếu chưa thì thêm tiếp vào nếu đã có rồi thì bỏ qua.

Hàm kiểm tra trùng lặp isDup sử dụng vòng lặp for duyệt theo từng bộ ba trong mảng result để kiểm tra từng phần tử tại vị trí i,j,k có trùng lặp không