# Opgave 1

I opgave 1 er den bedste måde at søge igennem træet på, post-order traversal. Det er det fordi, ved post-order traversal, søges nodernes børn igennem før noden selv bliver undersøgt. Dette gør at man hurtigt kan finde det største resultat. Da man ved at den højeste værdi er allermest til højre, kan man i princippet udelade funktionalitet at gå til venstre. Under er noget pseduo kode som beskriver post-order traversal

PostOrderTraversal(node):

hvis node er ikke null:

// Søg igennem venstre deltræ

PostOrderTraversal(node.venstre\_barn)

// Søg igennem højre deltræ

PostOrderTraversal(node.højre\_barn)

// Behandl nodeværdien

Udskriv(node.værdi)

# Opgave 2

Under er pseudokode for en funktion der sortere et array i omvendt rækkefølge

omvendtInsertionSort(array):

var n = array.længde();

for (var i = 1; i < n; i++):

var key = arr[i];

var j = i - 1;

imens (j >= 0 && arr[j] > key) {

arr[j - 1] = arr[j];

j = j + 1;

}

I programmet kan der ses at det sorteret array er 9 8 7 6 5 4 3 2.

For at sortere arterne med quicksort skal der vælges et pivotpunkt som resten af rækkefølgen bliver sorteret efter. Under kan der ses mit pivotpunkt.

Barracuda

Salamander

Nightingale

Capybara

**Elephant - pivotpunkt**

Gorilla

Under kan der ses hvordan processen foregår:

Et billede, der indeholder tekst, diagram, skærmbillede, Parallel

Automatisk genereret beskrivelse

Så ville man vælge et nyt pivotpunkt osv.

For at lede efter ordet Piratfisk i listen kan man bruge binary search der har en worst case på O(log(n))[[1]](#footnote-1). Dog skal man listen være sorteret så det vil se sådan ud med de samme dyr som før

Barracuda

Capybara

Elephant

Gorilla

Nightingale

Piratfisk

Salamander



1. <https://www.geeksforgeeks.org/binary-search/> [↑](#footnote-ref-1)