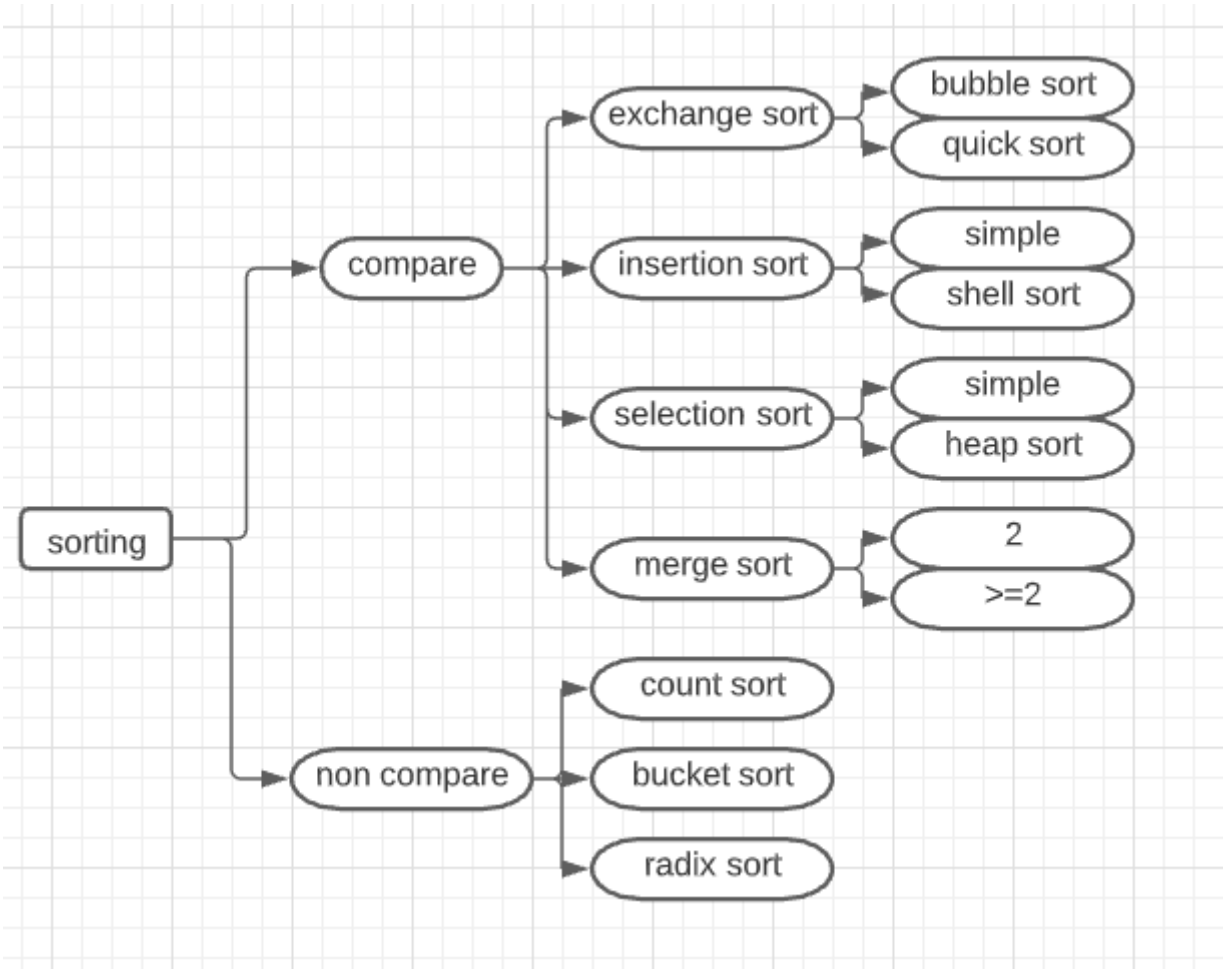
Algorithms - 3 Sorting

Summary of sorting algorithm

why sorting?
Sorting Problem
https://www.cnblogs.com/onepixel/p/7674659.html



Name	Average Time	Worst Time	Best Time	Space	Stability
Insertion Sort	$O(n^2)$	$O(n^2)$	O(n)	O(1)	Yes
Merge Sort	$O(nlog_2n)$	$O(nlog_2n)$	$O(nlog_2n)$	O(n)	Yes

Name	Average Time	Worst Time	Best Time	Space	Stability
Heap Sort					

Details of each sorting

1. Insertion Sort: O(n^2)

```
INSERTION-SORT(A):
    for j in range(2, len(A)): # after every loop A[:j] is sorted
        key = A[j]
        i = j - 1
        while i> 0 and A[i] > key:# find a suitable place for key
            A[i+1] = A[i]
            i = i-1
            A[i+1] = key
```

2. Merge Sort: O(nlogn)

```
MERGE-SORT(A,p,r):
    if p < r:  # check base case
        q = floor((p+r)/2)
        MERGE-SORT(A,p,q)
        MERGE-SORT(A,q+1,r)
        MERGE(A,p,q,r)

MERGE(A,p,q,r):
    n1 = q-p+1
    n2 = r-q
    let L[1...n1+1] and R[1..n2+1] be new arrays   # copy two original list
    for i = 1 to n1:
        L[i] = A[p+i-1]
    for j = 1 to n2:
        R[j]- A[q+j]
    L[n1+1] = infinite
    R[n2+1] = infinite
    for k = p to r: # put back into A in order
        if L[i] <= R[j]:
            A[k] = L[i]
            i = i+1
        else A[k] = R[j]
            j = j+1
```