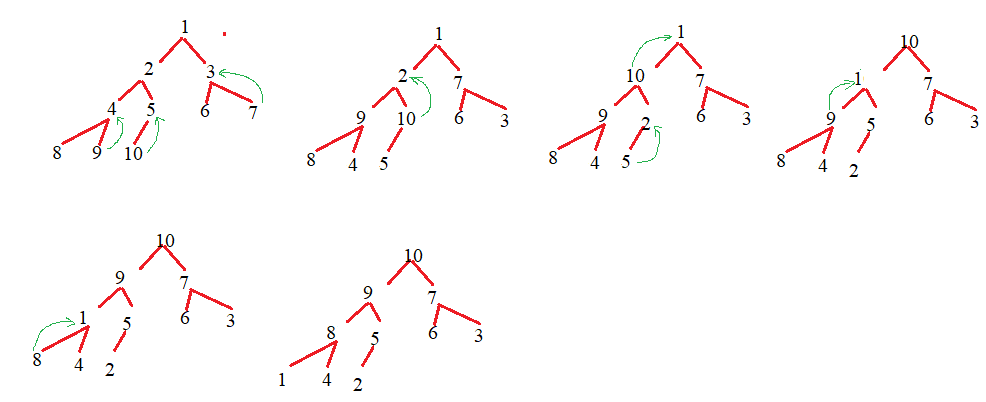
**Homework lecture 1**

**Sort**

2. Same question for these lists:

* 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
* Selection sort:
  + Assign min = first element (min = 1)
  + Compare min to 2nd element, 3rd element,... to get min element from 1st to last element (min = 1).
  + Cause min = first element so no swapping.
  + Assign min = 2nd element (min = 2)
  + Compare min to 3rd element, 4th element,... to get min element from 2nd to last element (min = 2).
  + Cause min = 2nd element so no swapping.
  + Do the same until reach the last element in array. (Number of comparison time: 9 + 8 + 7 + ... + 1 = 45, Number of swapping time: 0).
* Insertion sort:
  + Assign key = 2nd element (key = 2).
  + See that 2 > 1 so keep 2 stay same position.
  + Assign key = 3rd element (key = 3).
  + See that 3 > 2 so keep 3 stay same position.
  + Assign key = 4th element (key = 4).
  + See that 4 > 3 so keep 4 stay same position.
  + Do the same until reach the last element in array. (Number of comparison time: 9, Number of swapping time: 0).
* Bubble sort:
  + Compare 1 to 2, see that 1 < 2 so keep their position.
  + Compare 2 to 3, see that 2 < 3 so keep their position.
  + Compare 3 to 4, see that 3 < 4 so keep their position.
  + ...
  + Compare 9 to 10, see that 9 < 10 so keep their position.
  + Got 10 is the greatest element in array stay in last position.
  + Do the same until nothing left to compare. (Number of comparison time: 9 + 8 +... + 1 = 45, Number of swapping time: 0).
* Merge sort:
  + A red lines with numbers

    Description automatically generated with medium confidence
* Quick sort:
  + A diagram of numbers and points

    Description automatically generated
* Heap sort:
  + 
* Radix sort: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
  + Largest element in array is 10, which has 2 digits.
  + Sort array based on unit place:
    - 10, 1, 2, 3, 4, 5, 6, 7, 8, 9
  + Sort array based on tens place:
    - 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
  + Array after sorting:
    - 1, 2, 3, 4, 5, 6, 7, 8, 9, 10