

# Lecture 3: Heap Sort and Quick Sort

- Learn how performance of Merge sort can be further improved by using Heap sort and Quick Sort
- 

## Heap Sort

- Use "heap" data structure to manage information
  - Makes also an efficient priority queue (more later in the semester)
- 

## Binary Heap

pseudo-code for binary heap

```
Parent(i)
    return (i-1)/2
Left(i)
    return 2i+1
Right(i)
    return 2(i+1)
```

---

## Max-heap

- $A[\text{Parent}(i)] \leq A[i]$
  - Smallest value stored at root
  - Subtree rooted at node contains no values smaller than value of node itself
- 

## Max Heapify

```
l ← Left(i)
r ← Right(i)
if l ≤ heap-size[A] and A[l] < A[i]
    largest = l
else
    largest = r
if largest ≠ i
    then exchange A[i] ↔ largest
    MAX-HEAPIFY(A, largest)
```