Priority queue

- An abstract data type which allows elements to be arranged and accessed according to their respective priority values.
- The elements form a totally ordered set.
- When an element is inserted, the queue has to be rearranged with the minimum value being at the front of the queue
- Only smallest element can be deleted.
- Priority queue can be used for sorting.
- Build the queue in first phase
- Perform deletemin to output the element one after the another till the queue is empty.

Operations on priority queue

- Generic operations are `size()`, `isEmpty()`
- `insertItem(k,e)`: inserts element and its associated key value.
- `minElement()`: returns the element with minimum key value. Error if empty.
- `minKey()`: return the minimum key value. Error if empty.
- `removeMin()`: deletes the minimum element. Error if empty.

Implementation of priority queue

- Sequence allows operations of both List and Vector.
- Also has two mapping operations for bridging Vector operations with those of List.
- These are
  - `atRank(int r)`: returns postion of element at rank/index `r` and
  - `rankOf(Position p)`: returns the index/rank given a position.
- Either a sorted sequence or an unsorted sequence can be used.

Implementation using sequence

- Using an unsorted sequence
  - `insertItem` takes O(1) time, any of the insert operation can be used.
  - `minElement, minKey, removeMin` each takes O(n) time, we have scan the entire sequence using an iterator and then applying remove.
- By using a sorted sequence `removeMin` can be implemented as `S.remove(S.first())`, `S` stores the priority queue.
- But `insertItem` has to scan the sequence and insert the item only at position such that the sequence remains sorted. In the worst case takes time O(n).
- Heap seems to be most elegant way to implement a priority queue. The heap implementation is discussed next.

Heap

- Special kind of complete binary tree.
- Stores elements from a totally ordered set.
- Element at any internal node is smaller in value than those of the elements stored at any of its two children.
- Also known as a priority queue.
- Has many applications.
- The minimum number of nodes in a heap of height h is 2^{h-1} and maximum number of nodes is $2^h - 1$
- This means if n is total number of nodes in a heap then $n \leq 2^h - 1$ and $2^{h-1} \leq n$.
- Therefore, $h \leq \log n + 1$ and $\log(n+1) \leq h$, i.e., $h = \lceil \log(n+1) \rceil$.