

Dynamic Sets and Data Structures

- Over the course of an algorithm's execution, an algorithm may maintain a dynamic set of objects
- The algorithm will perform operations on this set
 - Queries
 - Modifying operations
- We must choose a data structure to implement the dynamic set efficiently
- The “correct” data structure to choose is based on
 - Which operations need to be supported
 - How frequently each operation will be executed

Some Example Operations

- Notation
 - S is the data structure
 - k is the key of the item
 - x is a pointer to the item
- $\text{Search}(S,k)$: returns pointer to item
- $\text{Insert}(S,x)$
- $\text{Delete}(S,x)$: note we are given a pointer to item
- $\text{Minimum or Maximum}(S)$: returns pointer
- $\text{Decrease-key}(S,x)$
- $\text{Successor or Predecessor}(S,x)$: returns pointer
- $\text{Merge}(S_1,S_2)$

Basic Data Structures/Containers

- Unsorted Arrays
- Sorted Array
- Unsorted linked list
- Sorted linked list
- Stack
- Queue
- Heap