

d\_store.h  
d\_.h  
d\_.h

**EECE 2560: Fundamentals of Engineering Algorithms**  
Department of Electrical and Computer Engineering

**Project Word Search**

**Part b**

In this part, solve the word search puzzle by using the `quicksort` algorithm and the `heapsort` algorithm to sort the words in the dictionary.

1. Implement a member function of `dictionary` that sorts the words using `quicksort`
2. Implement the template class `heap<T>` that stores objects in a heap of type `vector<T>`, and which includes:
  - (a) functions `parent(int)`, `left(int)`, `right(int)`, and `getItem(int n)` which returns the `n`th item in the heap, **return T**
  - (b) for a max-heap, functions `initializeMaxHeap()`, `maxHeapify()`, and `buildMaxHeap()`,
  - (c) function `heapsort()`.

3. Implement a member function of `dictionary` that sorts the words using `heapsort`.

Since the heap is only used to sort the word list, you can declare the heap within the `dictionary::heapsort` function, copy the unsorted words into the heap, sort the words, and then copy the words out. Then the `dictionary::binarySearch` function can be used to look up words. **\* selectionsort, quicksort, heapsort**

4. Modify the global function `search(int)` which reads the name of the grid file from the keyboard and prints out all words from the word list that can be found in the grid. The integer parameter is used to select the sorting algorithm used.

**initialize maxHeap(vector<T> values) (words in dictionary obj)**

- (1) copy values to data member items (heapsize)**
- (2) buildMaxHeap() { call maxHeapify() }**
- (3)**

**vector<T> heapSort(); // return the sorted items**

**l → later, copy back to words in dictionary**