Hash Table and Generic ADT

Project 3

Generic ADT (2nd week)

- Change directory to “generic”
- Create table.c to implement the set operations with generic data type

```
struct set {
        int count;
        int length;
        void **data;
        char *flags;
        int (*compare)();
        unsigned (*hash)();
        };
```

```
SET *createSet(int maxElts, int (*compare)(), unsigned
(*hash)());

static int search(SET *sp, void *elt, bool *found)

void addElement(SET *sp, void *elt); (does not allocate new memory)

void removeElement(SET *sp, void *elt);

void *findElement(SET *sp, void *elt);

void *getElements(SET *sp);
```

Changes

- Our new generic set ADT does not know the target data type, so we do not know how much memory we want allocate: no strdup.
- Deallocate the memory that only you allocated: no free in removeElement. What about the destroySet?
- Change char * to void *
- Function pointer

Function Pointer

```
SET *createSet(int maxElts, int (*compare)(), unsigned (*hash)()):
      ……
      sp→compare = compare;
      sp→hash = hash;
      ……
static int search(SET *sp, void *elt, bool *found):
      ……
      strhash(elt) ⟶ (*sp→hash)(elt)
      strcmp(sp→data[locn], elt) ⟶ (*sp→compare)(sp→data[locn], elt)
      ……
```

Test Cases

- `./unique /scratch/coen12/Macbeth.txt`
- `./unique /scratch/coen12/Macbeth.txt /scratch/coen12/Bible.txt`
- `./unique -l /scratch/coen12/Macbeth.txt /scratch/coen12/Bible.txt`
- `./parity /scratch/coen12/Macbeth.txt`
- `./counts /scratch/coen12/Macbeth.txt`

Submission

- Report.txt
- Big O Time Complexity for Each Function
- File:
  - tar -czvf project3.tar folder_path
    - folder_path is the directory of the folder that contains both “strings” folder and “generic” folder
- Demo deadline: the end of next lab session.