

## Project #2

---

### Dropbox-like Storage

Create your own dropbox-like data storage system. Your storage system should accept:

- *loading of text files*: When a file is loaded, it is divided into fixed sized chunks (e.g. 1K), the last chunk is padded such that it has the same size with the other chunks, the chunks are stored into your master storage unless they have not been stored before, and the filename to chunk relation is marked
- *listing of loaded files*: Your system should be able to list the names of the existing files
- *retrieving back of text files*: When a request to download a file is done, it should be generated back from its chunks and returned back to the user as it was uploaded
- *deleting of text files*: when a file delete is requested, the marks associating the file with the chunks and the chunks associated only with this file must be removed.

For example, consider that first you first received and stored file *f1* comprising of chunks [*c1*, *c2*, *c3*] and now you receive file *f2* comprising of chunks [*c2*, *c3*, *c4*]. Assuming that each chunk size is 1K, your system should only store around 4K for keeping the chunks along with some storage used for keeping the file names. Note that if file *f1* is deleted from your system now, your storage space will reduce only 1K as you can remove only the chunk storage associated with *c1*.

You are free to use any data structure you like for storing the chunks, the file names, etc... But you must define the complexity of each functionality in your storage system, e.g. loading the files, listing file names, retrieving the files, deleting files, etc... Efficient data structure selection & usage will get higher points.

### Minimum Requirements [55%]

- Command line user interface that lets inserting, listing, retrieving, deleting files.
  - Ability to store ten 10MB ASCII files, any two of which differ in at most 5 character edits<sup>1</sup>, using at most 15MB of storage.
  - Ability to retrieve all files stored in any order at any time.
- 

### Possible Extensions [15%]

- Develop a GUI [15%]
- 

### Comparative Features [30%]

Your algorithms and data structures will be compared against others in the class and ranked according to:

- Reductions obtained in storage space
- Speed of key operations

---

<sup>1</sup> Character edits are insertion, deletion, or modification of a character.