Internals Project #2 Documentation

Chen Zou, Jiaqi Liu,
czou@wpi.edu, jliu6@wpi.edu

- Assumption:
  - Based on the requirement, we need create three interfaces to operate the index and implement two interfaces to return the data required.
  - All inputs of the hash function are regarded as string.
  - We assume rids are unique.
- Design decision:
  - We avoid using built-in type dictionary because it is implemented using hashing itself.
  - When duplicated data_value emerges, we store it in a list.
  - First read the file. We download the data from the website and stored the data in file (data.txt).
  - Secondly although the project require us to only set up two interfaces, which are given the year or year and format data program should use index to return the data based on the requirement, we set a customized interface that is our program can set up any single attribute or combined attributes based on the raw data. That means, if we want to set an index based on the attributes writer and country, our program can easily deal with that.
  - Thirdly we think our key point is to design the hash function
    - Common string hash function includes BKDRHash, APHash and DJBHash , etc. There are two reasons to choose BKDRHash:
      - 1. It performs best in hashing random string composed of English letters and digits and semantical English words. (By performing best, we mean quick calculation and low collision rate.)
      - 2. It is easy to implement.The implement of BKDRHash is quite simple. We need a seed, which can be arbitrarily chosen from a set of candidates.
    - For each character in a string, we re-caculate the hash value based on it. This procedure renders BKDRHash ability to handle different sizes of strings.
    - Since there are approximately 40 records in the file, 8 buckets should be fair enough. As the result, we use only last three digits of the binary hash value BKDRHash produce.
  - To test our program, we set several test cases.