CSC553 Assignment 2
Kaijun He

1. Consider the following dataset:

   ['bab3', 'bc01', 'cc2', 'cd5', 'cd3', 'cdx2', 'cdx1', 'e01', 'g02', 'ha1', 'hb1', 'hc8', 'hz5', 'z00', 'z01', 'bc01']
   a) This hashing function is not a good hashing function, because the hashing function distributed the dataset not equally. For examples, for bucket3 it only contains one data, and for bucket 4, it contains 8 dataset, so the distribution is really bad, I will say this hashing function is bad.
   b) For my design, I will design the hash function into 5 bucket as below.
      [only first letter] -> bucket1{'e01', 'g02', 'z00', 'z01'}
      [2 letter & first letter is b]->bucket2{'bc01', 'bc01' }
      [2 letter & first letter is c] -> bucket3{'cc2', 'cd5', 'cd3'}
      [2 letter & first letter is h] ->bucket4{'ha1', 'hb1', 'hc8', 'hz5'}
      [3 letter]->bucket 5{'bab3','cdx2', 'cdx1'}
   c) [first letter is a & number is 1]->bucket1{ a1', 'a1', a1', 'a1', a1'}
      [first letter is b-c & number is 1]->bucket2{ b1', b1', 'c1', 'c1', 'c1'}
      [first letter is d | number is 2]-> bucket3{'d2','d1', 'd2' , 'a2'}
2.
3. Consider a relation r that takes up 155 blocks on disk. If you have exactly 10 blocks of memory available:
   a) There exactly 10 blocks of memory, we have an initial sorting phase, so in total of 155 block we will divide all blocks into 16 runs. We know M = 10, each pas can merges 9 runs. For merge pass-1 run into two, one is 90 blocks, one is 65 and then do merge pass-2 merge into one, therefore there are 2 merge pass and 1 sorting pass; so 3 pass in total.
   b) Since we know the merge pass number is 2 and blocks size is 155, therefore by formula is 2 * 155 + 2*155*2 = 930.
   c) 155/M <= M-1 then solve the equation, we get M >= 12.95 so M should be >= 13
   d) We know M = 8 , therefore for 3 passes, we have M- 1 = 7, for exactly 3 passes in total, we will have 2 * 7 * 8 =112, the biggest data can relation r get is 112 blocks on disk

4. Consider two relations r (200 blocks) and s (30 blocks)
   a) We have br = 200 blocks, bs = 30 blocks, M = 8, by formula, we have bs*[br/(M-2)] + br transfer = 30 *[200 /6] + 200= 1220 for transfer, and seeks 2*[br/(M – 2)] = 2 *[200 /6] = 68 for seeking
   b) We have br = 30 blocks, bs = 200 blocks, M = 8, by formula, we have bs*[br/(M-2)] + br transfer = 200 *[30/6] + 30 = 1030 for transfer, and seeks 2*[br/(M – 2)] = 2 *[30 /6] =10 for seeking. Therefor reversed br and bs, result for transfer and seeking is less than results in part a).
   c) If we have M = 35, then M-2 = 33, br = 200. Bs = 30, therefore by formula = 30*[200/33] + 200 = 410 for transfer, and 2 * [200/33] = 2 * 7 = 14 for seeking.
5. B-Trees
   a) 8*n + (n - 1)*8 <= 512
      16n <= 520
      Therefore n = 32, the order is 32
   b) 12 * n + (n -1)*8 <=512
      20n <= 520

n = 26, therefore new order will be 26

c) We have order 32,
For level 0 : 1- 31
For level 1 :32 – 1023
For level 2 : 1024 – 32767
For level 3 : 32768 – 1048575
For level 4 : 1048575 – 33554431
We have 4200000 index, therefore we have height of tree is 4.

d) 4200000 / 33554431 = 0.125169757775 = 12.52%

e) For this problem , in part c, I list level 4, maximum number of values I can index at total of 4 level which root plus 3 more level