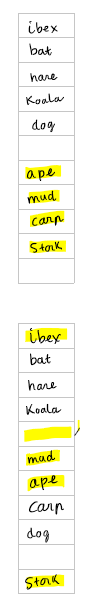
## Exercise 1

## 



## Exercise 2

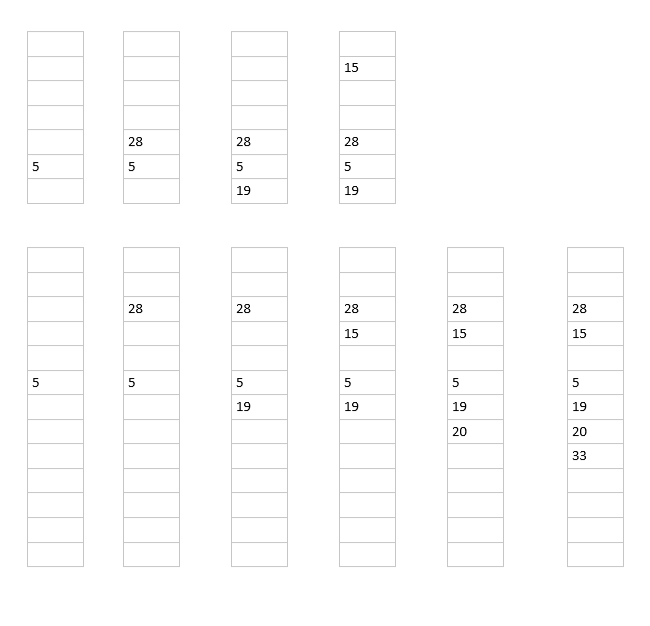
a) Worst case: h(k) give same result for all keys, as a result, they are stores as an unsorted linked list

Search time 0(n)

Example: table size 10, key = {2,22,32,42,62}

b) Because in worst case scenario, search time is O(n), much higher than O(1), this hash table is not suitable for a time-critical application

## Exercise 3



## Exercise 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Insert | Retrieve | Delete |  |
| Array | n | 1 | n | Slow to insert and delete, very fast to retrieve |
| Linked List | n | n | n | Slowest algorithm |
| Balanced Binary tree | Height of tree | Height of tree | Height of tree | Time complexity reduce as we have more variables to track |
| Hash Table | 1 – n (depend on hash function and collision handling) | 1 – n (depend on hash function and collision handling) | 1 – n (depend on hash function and collision handling) | Fastest algorithm  Can become slower depend on hash function, table size, collision handling |