Question 1:

Suppose we need to store a set of 2000 items for which the keys are character strings of length 9. Each key consists of a 9-character combination of the characters "X", "O" and "E". For example, some possible keys are "EOXEEEEEX" , "EXEXEXOOX", "OOXXOOOOO" etc.

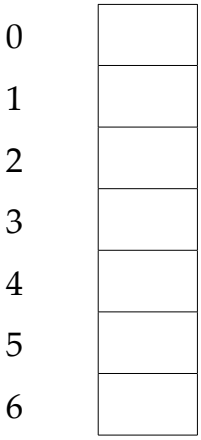
Choose a table size, a hashing function, and a form of open addressing for collision resolution (give as much detail as you can). Explain your choices.

Hint for 2020 students: the first step is to convert the keys into integers, bearing in mind that many keys will have the same distribution of letters (such as 4 X's, 3 E's and 1 O)

Question 2

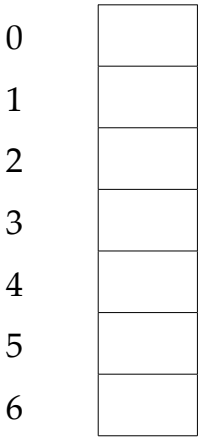
Consider a hash table of size 7. Draw the table that results after inserting the values 13, 35, 27, 8, 14, 17 in that order …

(a) when collisions are resolved by chaining, using h(k) = k mod 7



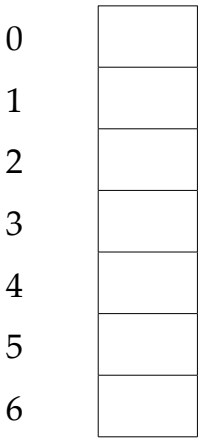
(b) when collisions are resolved by linear probing, using

$$h(k,i) = (k + i) \bmod 7$$



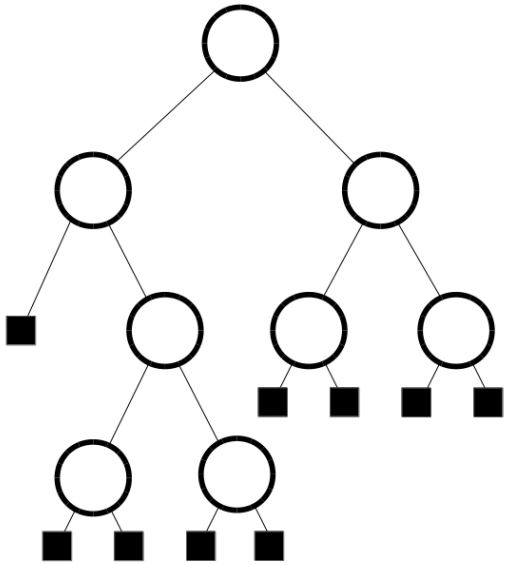
(c) when collisions are resolved by double hashing, using

$$h(k,i) = (k + i*(1+(k \bmod 5))) \bmod 7$$



Question 3:

Here is a binary tree. The square vertices represent empty leaf-nodes. Either explain why this tree cannot be legally coloured as a Red-Black Tree, or give it a legal Red-Black colouring.



Question 4:

Here is a collision resolution method that combines chaining with open addressing. In this method we resolve collisions with chaining, but each chain is allowed to contain at most 2 values. If we try to insert a value into a location where the chain is full, we use linear probing to find the next address where the chain has length < 2.

a) [4 marks] Show the result of inserting these values :

3, 23, 14, 6, 13, 24, 25, 17, 4

into this hash table using h(k) = k % 10

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

(This question continues on the next page.)

b) [4 marks] How would you handle the "delete" operation, so as to guarantee that "search" would function properly?

c) [4 marks] Compared to standard open addressing with linear probing, would you expect this method to have more primary clustering, less primary clustering, or about the same amount of primary clustering? Explain your answer.