Name: Maahir Kalban (cvt547)

CS5463 Fundamentals of Software

Homework 3

Due 10/30/20 before 11:59pm (Central Time)

1. Arrays and Heaps (6 points)

(1) (3 points) Suppose you wanted to create an algorithm to find the number of elements $\geq k$ in an array, A, of n distinct elements.

Specifically, describe efficient algorithms to solve this problem when A is the following data structures.

- (a) Unsorted array
- (b) Sorted array
- (c) Max-heap

(2) (3 points) Analyze the runtime of your three algorithms from the previous part:

- (a) Unsorted array
- (b) Sorted array
- (c) Max-heap

2. Huffman Encoding (2 points)

(1) Show the process by which a Huffman tree would be built for the following string of character: (i.e., redraw the tree after each new node is created).

"cannercancanacanra"

(2) Using your tree, find the encoding that would result for the above string.

3. Red-Black Trees (2 points)

(1) Company X has created a new variant on red-black trees which also uses blue as a color for the nodes. They call these "red-black-blue trees". Below are the new rules for these trees:

- Every node is red, blue, or black.
- The root is black.
- Every leaf (NIL) is black.
- If a node is red, then both its children are black.
- If a node is blue, then both its children are red or black.
- For each node, all simple paths from the node to descendant leaves contain the same number of black nodes.

(a) (2 points) In class we found that the height, h, of a red-black tree is $\leq 2\log_2(n+1)$ (where n is the number of keys). Find and prove that a similar bound on height of the red-black-blue trees.

(**Hint:** You can use the same approach as we did to show

$h \leq 2\log_2(n+1))$.

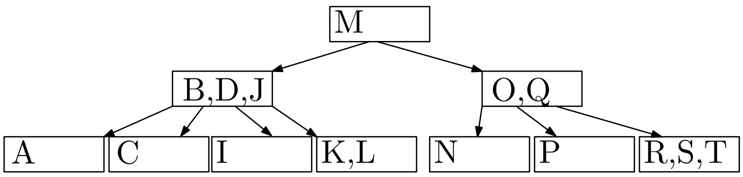
(b) (0 points - just for fun) Adding an additional color didn't seem to improve our bound on h (i.e., 3 colors allows the tree to become more unbalanced than with 2 colors). What benefit might we get from the extra color?

4. B-trees (4 points)

(1) (2 points) Show the results of inserting the keys

E, F, G, U, V, W, H

in order into the B-tree shown below. Assume this B-tree has minimum degree $k = 2$. Draw only the configurations of the tree just before some node(s) must split, and also draw the final configuration.



(2) (2 points) Suppose you have a B-tree of height h and minimum degree k. What is the largest number of keys that can be stored in such a B-tree? Prove that your answer is correct.

(**Hint:** Your answer should depend on k and h. This is similar to theorem we proved in the B-tree notes).

5. Hash Table Probabilities (3 points)

(1) (1 point) Suppose 2 keys are inserted into an empty hash table with m slots. Assuming simple uniform hashing, what is the probability of:
(a) exactly 0 collisions occurring
(b) exactly 1 collisions occurring

(2) (2 points) Suppose 3 keys are inserted into an empty hash table with m slots. Assuming simple uniform hashing, what is the probability of:
(a) exactly 0 collisions occurring
(b) exactly 1 collisions occurring
(c) exactly 2 collisions occurring

6. Hash Table (7 points)

(1) Consider inserting the keys $2, 21, 3, 58, 11, 42, 34$ into a hash table of length $m = 10$ with the hash function $h(k) = k \bmod 10$.

(a) (2 points) Illustrate the result of inserting these keys using linear probing to resolve collisions.
(b) (2 points) Illustrate the result of inserting these keys using chaining to resolve collisions.

(2) Consider inserting the keys 8, 5, 14 into a hash table of length $m = 8$ with the hash function $h(k) = \lfloor m(kA - \lfloor kA \rfloor) \rfloor$ where $A = 0.625$.
(a) (2 points) Illustrate the result of inserting these keys.
(b) (1 point) Now compute the hash function of the key 14 using the alternative algorithm described below.

You can assume we have a word size $w = 4$. Since $m = 8 = 2^3$, $p = 3$. Since $A = 0.625 = 10/2^4 = 10/2^w$, $s = 10$.

Alternative Algorithm: Compute ks and convert it to a binary number. This number will consist of $\leq 2w$ bits. Look at the rightmost w bits. Of those bits, convert the leftmost p bits back to an integer. This integer is your hash table slot.

1) Given Array A with n elements, to find number of elements >= k.

a) unsorted array, iterate through all the elements keeping track of elements >= k
Thus complexity is $\underline{O(n)}$.

b) Sorted array, use binary search to find elements >= k. Complexity $\underline{O(\log n)}$

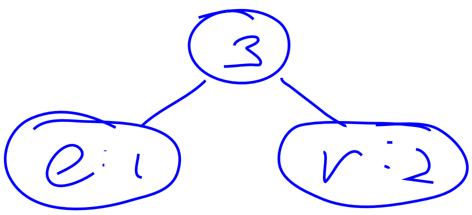
c) Max heap, Always returns maximum element.
Removing and heapifying operation takes $O(\log n)$.
We also need to remove elements >= k.
Thus total complexity is $\underline{O(n\log n)}$

2) a) String : cannercancanacan ra

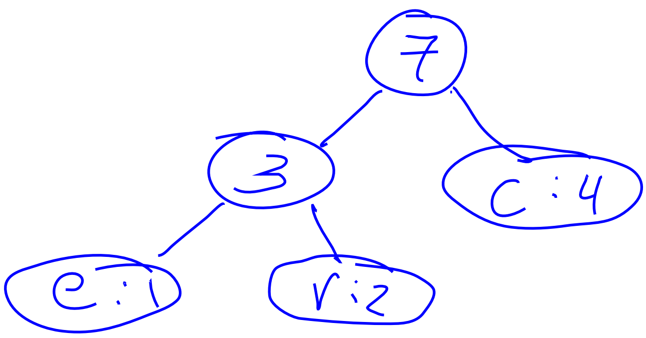
Character	frequency
a	6
c	4
e	1
n	5
r	2

Huffman tree with 5 nodes (a, c, e, n, r)

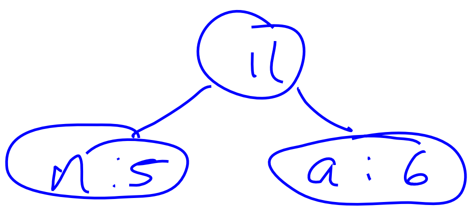
- choose nodes (e and r) lowest frequency



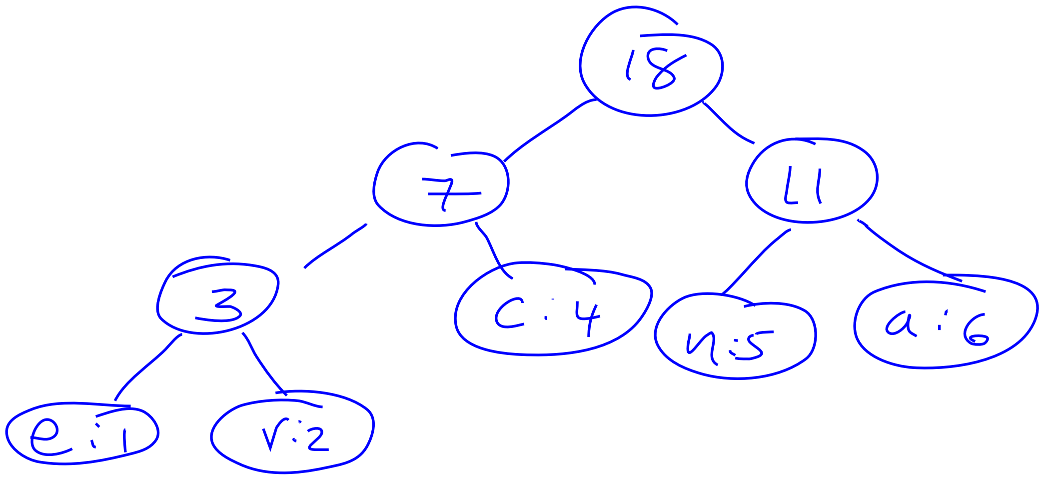
- choose nodes (e+r) and c



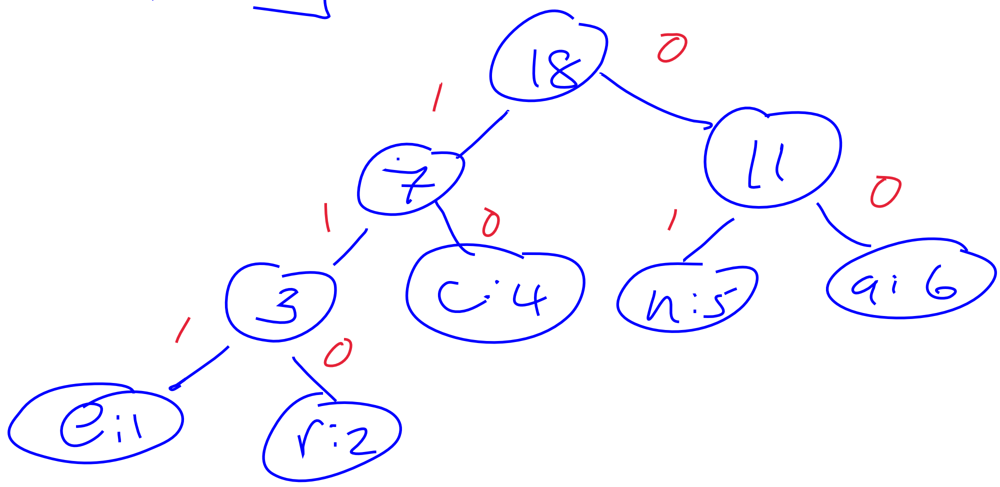
- choose node a and n



\- Final node



b) Encoding



Character	Code
a	00
c	10
e	111
n	01
r	110

Encoding for cannercancanacanra is

10 00 01 01 111 110 10 00 01 10 00 01 00 10 00 01 110 00

3.) a) To prove $h \leq 2\log(n+1)$

where h is the height of the tree an n is the number of keys.

Lemma: A subtree of height $bh(v)$ in a red tree contains atleast $2^{bh(v)} - 1$ internal vertices.

$$n \geq 2^{bh(v)} - 1$$

Proof: Let P be a path in the red-blue-black tree

Atleast half of the nodes on the path must be black since every blue and red node is followed by black node.

Hence, the black height of the root is atleast $h/2$.

We have, $n \geq 2^{bh(v)} - 1$

$$n \geq 2^{h/2} - 1$$

$$\Rightarrow n+1 \geq 2^{h/2}$$

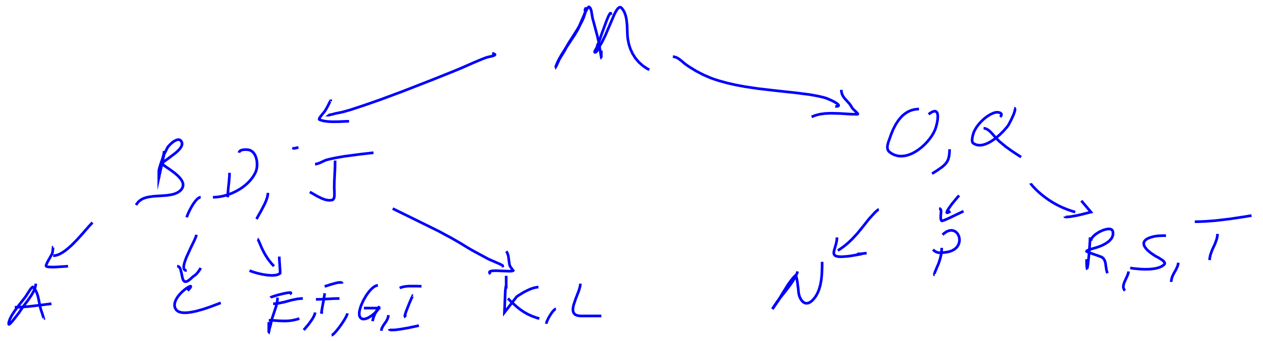
$$\log_2(n+1) \geq \log_2(2^{h/2})$$

$$\Rightarrow \log_2(n+1) \geq h/2$$

Thus, $\Rightarrow$ $\underline{h \leq 2\log_2(n+1)}$

4 a) Atmost $2t = 2(2) = 4$ children, $2t - 1 = 2(2) - 1 = 3$ keys

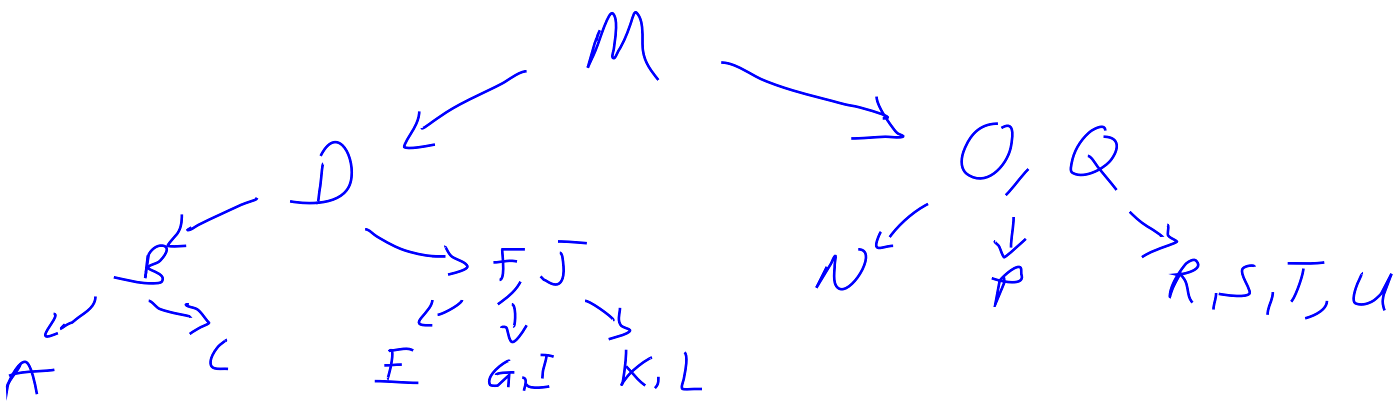
insert E, F, G



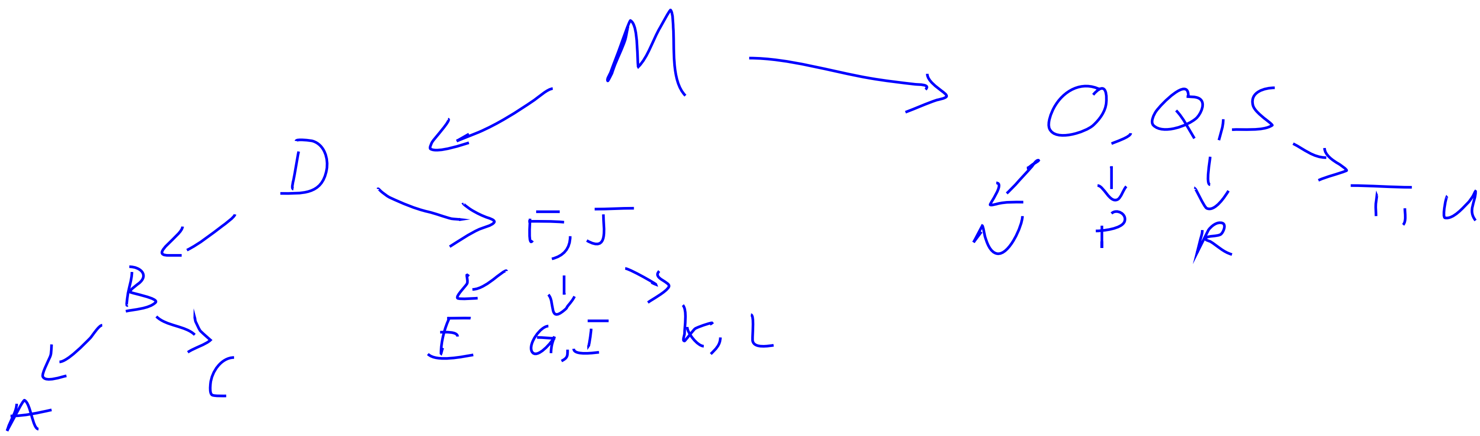
breakdown

breakdown

insert U

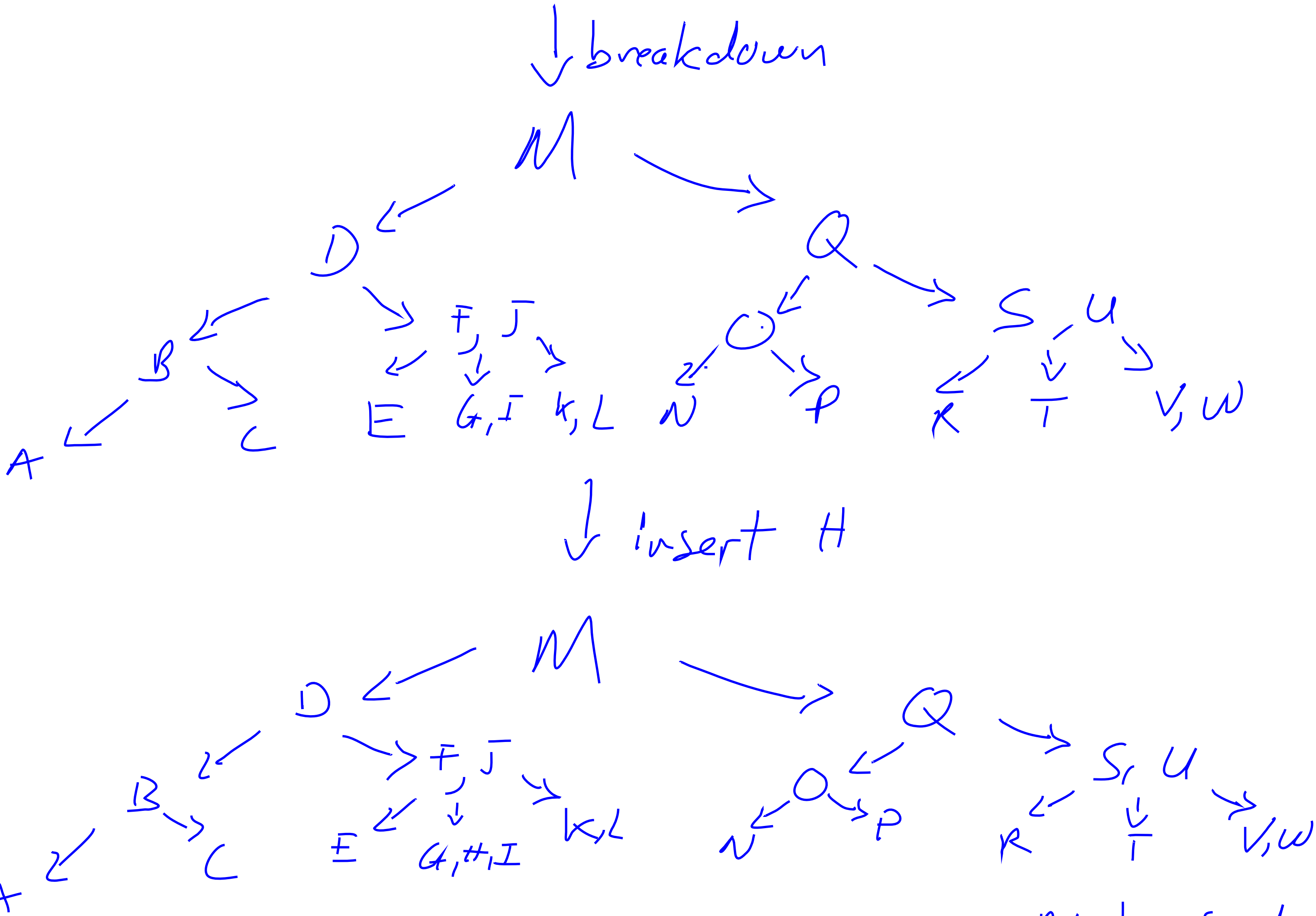


breakdown

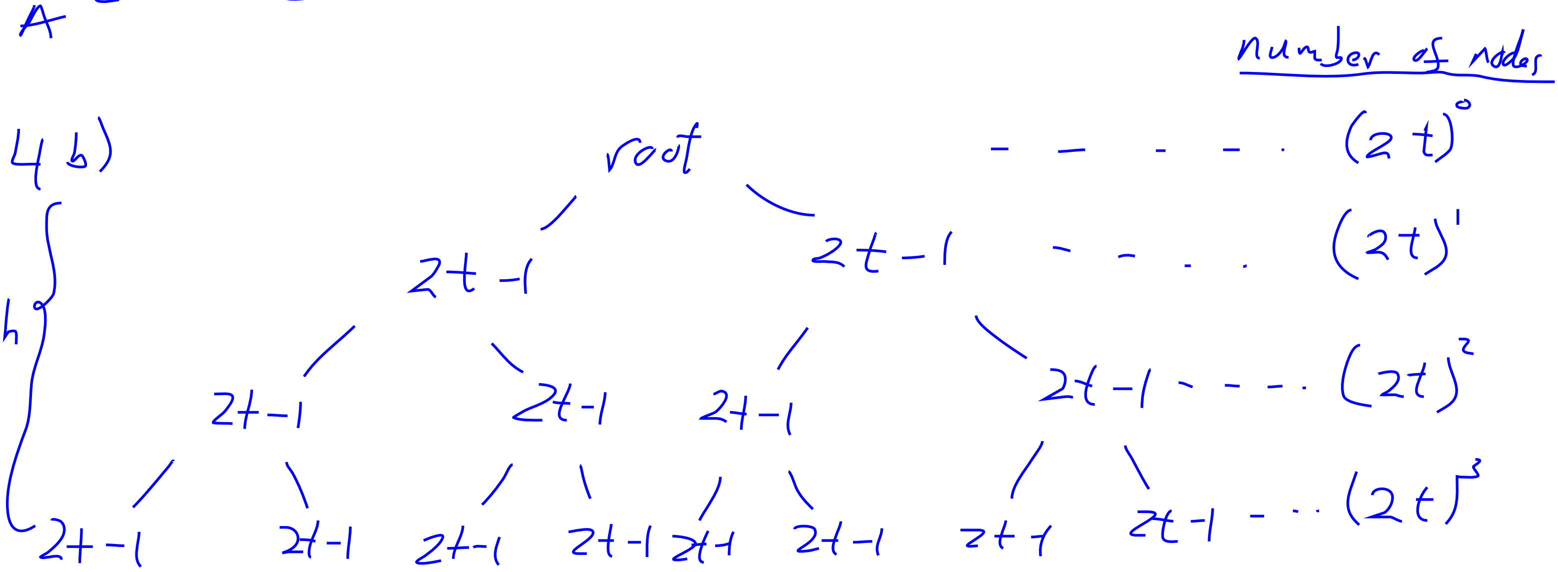


insert V, W

breakdown



4 b)



→ each node contains 2t-1 keys

→ height = h.

→ so there is atmost $(2t)^k$ nodes

Total nodes will be the sum of

$(2t)^0, (2t)^1, (2t)^2, \ldots, (2t)^h$

Maximum number of keys

$$= (2t-1)\left[(2t)^0 + (2t)^1 + (2t)^2 + \dots + (2t)^h\right]$$

$$= (2t-1) \sum_{i=0}^{h} (2t)^i$$

Geometric series

$$= (2t-1) \frac{2t^{h+1} - 1}{(2t-1)}$$

$$= \underline{2t^{h+1} - 1}$$

5. (1)

a) $P(X=x) = \binom{m}{x} p^x (1-p)^{m-x}$

here, $p = \frac{1}{2}$

$P(X=0) = \binom{m}{0}\left(\frac{1}{2}\right)^0 \left(1-\frac{1}{2}\right)^{m-0} = \left(\frac{1}{2}\right)^m$

b) $P(X=1) = \binom{m}{1}\left(\frac{1}{2}\right)^1 \left(1-\frac{1}{2}\right)^{m-1} = m \times \left(\frac{1}{2}\right)^{m-1}$

(2) $p = \frac{1}{3}$

a) $P(X=0) = \binom{m}{0}\left(\frac{1}{3}\right)^0 \left(1-\frac{1}{3}\right)^{m-0} = \left(\frac{2}{3}\right)^m$

b) $P(X=1) = \binom{m}{1}\left(\frac{1}{3}\right)^1 \left(1-\frac{1}{3}\right)^{m-1} = \binom{m}{1}\left(\frac{1}{3}\right)\left(\frac{2}{3}\right)^{m-1}$

c) $P(X=2) = \binom{m}{2}\left(\frac{1}{3}\right)^2 \left(1-\frac{1}{3}\right)^{m-2} = \binom{m}{2}\left(\frac{1}{3}\right)\left(\frac{2}{3}\right)^{m-2}$

6.

(1) a) Hash table size $m = 10$.

0	
1	21
2	2
3	3
4	11
5	42
6	34
7	
8	58
9	

$\rightarrow h(2) = 2 \bmod 10 = 2$

$\rightarrow h(21) = 21 \bmod 10 = 1$

$\rightarrow h(3) = 3 \bmod 10 = 3$

$\rightarrow h(58) = 58 \bmod 10 = 8$

$\rightarrow h(11) = 11 \bmod 10 = 1$ (collision)

check $1 + 1 = 2$ (collision)

$1 + 2 = 3$ (collision)

$1 + 3 = 4$ (place it)

$\rightarrow h(42) = 42 \bmod 10 = 2$ (collision)

check $2 + 1 = 3$ (collision)

$2 + 2 = 4$ (collision)

$2 + 3 = 5$ (place it)

$\rightarrow h(34) = 34 \bmod 10 = 4$ (collision)

check $4 + 1 = 5$ (collision)

$4 + 2 = 6$ (place it)

b)

0		
1	21	$\rightarrow$ 11
2	2	$\rightarrow$ 42
3	3	
4	34	
5		
6		
7		
8	58	
9		

$\rightarrow h(2) = 2 \bmod 10 = 2$

$\rightarrow h(21) = 21 \bmod 10 = 1$

$\rightarrow h(3) = 3 \bmod 10 = 3$

$\rightarrow h(58) = 58 \bmod 10 = 8$

$\rightarrow h(11) = 11 \bmod 10 = 1$ (chain to 21)

$\rightarrow h(42) = 42 \bmod 10 = 2$ (chain to 2)

$\rightarrow h(34) = 34 \bmod 10 = 4$

2) a) keys $= [8, 5, 14]$

$m = 8$, $A = 0.625$

function $h(k) = \lfloor m(kA - \lfloor kA \rfloor) \rfloor$

0	8
1	5
2	
3	
4	
5	
6	14
7	

$\rightarrow h(8) = \lfloor 8\,(8 \times 0.625 - \lfloor 8 \times 0.625 \rfloor) \rfloor$

$= \lfloor 8(5 - \lfloor 5 \rfloor) \rfloor$

$= \lfloor 8(0) \rfloor = 0$

$\rightarrow h(5) = \lfloor 8\,(5 \times 0.0625 - \lfloor 5 \times 0.625 \rfloor) \rfloor$

$= \lfloor 8\,(3.125 - 3) \rfloor$

$= \lfloor 8\,(0.125) \rfloor = 1$

$\rightarrow h(14) = \lfloor 8\,(14 \times 0.625 - \lfloor 14 \times 0.625 \rfloor) \rfloor$

$= \lfloor 8(8.75 - 8) \rfloor$

$= 6$

b) $p = 3$, $w = 4$

$s = 10$

compute $k \times s = 14 \times 10 = 140$

convert to binary $= 1000\ 1100 \leq 2w$

right most w bits, i.e. 4 bits

1100

convert the left most p bits to decimal i.e. 3 bits

$\underbrace{110}_{p}0$

$(110)_2 \rightarrow \underline{6_{10}}$

Thus the hash table slot for key 14 is $\underline{\underline{6}}$