CS 5200 Spring 2018 Analysis of Algorithms-Homework5

Xiongming Dai

April 13,2018

1 Comparisons among lists

Singly linked lists contain nodes which have a data field as well as 'next' field, which points to the next node in line of nodes. Operations that can be performed on singly linked lists include insertion, deletion and traversal,successor,predecessor,minimum and maximum, For singly linked list, including the unsorted and the sorted, we could find the running time for this operations:

Search(L,k):we would like to search a word for the length of the list n, the time is $..(n)$. Insert(L,x):if we would like to insert a word for the length of the list n, we just need one step, so the time run is $..1$. Delete(L,x):if we would like to delete a word x,firstly,we should to search this element, therefore, the time run is $..n$. Successor(L,x):For unsorted linked, we should firstly search all the elements, the time run is $..n$. For sorted linked, we do not need to search, therefore, the time run is $..1$. Predecessor(L,x):we should first search the element x, therefore, the time run is $..n$. Mninimum(L):For the unsorted, we need to find one by one, therefore, it is $..n$, for the sorted, it is only $..1$. Maximum(L):We need to search one by one for comparison, therefore, it is $...n$.

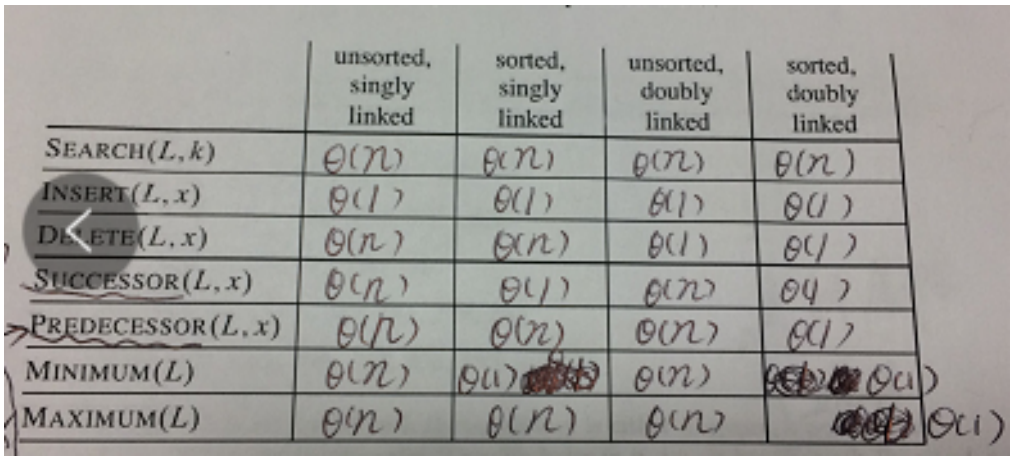
In a 'doubly linked list', each node contains, besides the next-node link, a second link field pointing to the 'previous' node in the sequence. The two links may be called 'forward('s') and 'backwards', or 'next' and 'prev'('previous').

Search(L,k):generally, the search is conducted one by one, so the time is $..n$. Insert(L,x):we just need to insert to the position where the value is x, the time is $..1$. Delete(L,x):For double linked list, there are two pointer, therefore, if we want to x, we just one step.The time is $..x$. Successor(L,x):For unsorted doubly linked, the time is $..n$. For sorted linked, the time is $..1$. Predecessor(L,x):For the unsorted, we should search the value x in the list, therefore, the time is $..n$, but for sorted list, it is only $..1$. Minimum(L):This opearation is similar to the operation of predecessor. Therefore, for the sorted, the time is $..n$. for the unsorted, the time is $..1$. Maximum(L):This opeartion is similar to the minimum, similarly, we can find the time needed is $..n$ for the unsorted, and $..1$ for the sorted.

The table is shown in Figure 1.

2 Hash table

According to hash function $h(k) = kmod9$, we can find the associated lable number as follows:
0: null
1:10,19,28
2:20



	unsorted, singly linked	sorted, singly linked	unsorted, doubly linked	sorted, doubly linked
SEARCH(L, k)	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
INSERT(L, x)	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$
DELETE(L, x)	$\Theta(n)$	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$
SUCCESSOR(L, x)	$\Theta(n)$	$\Theta(1)$	$\Theta(n)$	$\Theta(1)$
PREDECESSOR(L, x)	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(1)$
MINIMUM(L)	$\Theta(n)$	$\Theta(1)$	$\Theta(n)$	$\Theta(1)$
MAXIMUM(L)	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(1)$

Figure 1: lists

3:12
4:null
5:5
6:33,15
7:null
8:17

The left column is the label number, the back is the associated keys. From the talbe, we can find there are some collisions.

3 Auxiliary hash function

The total process of inserting these keys using linear probing, please see the attached following figure tabling with the hand-written. It is shown in figure....and figure.....

Secondary method using quadratic probing , similary, we can do the procedure as the following figure... and figure....

For the third method, similary, we have the figure .. and figure .....

4 Draw binary search tree

First, we could confirm the root of the tree. Usually, we could choose the center value 10 is choosed and establish the tree, then find the rule of the tree to confirm other trees with different height of binary search tree.

For the heights 2, the root is 10, for left subtree, the range is $[1, 10]$, the center is 4, then the left leaves is 1, and right is 5. Similary, for the right subtree, the range is $[10, 21]$, the center is 17, and the left is 16, the right is 21. Therefore, the structure with height 2 is finished. It is shown in figure ....

For the height 3, similary, we can change the structue of the right subtree based on the structure of height 2,we have the figure ....

Similary, we can build the trees with heights 4,5, and 6 using the same method. They are shown as following figure ....

5 Binary-search-tree and min-heap

From the definition of binary-search-tree, it is very easy for us to know,the left child of the tree is always smaller that the root and the right child of the tree is always larger than the root. For heap no matter max-heap or min-heap, it guarantees the children larger or smaller than the parents, while they did not know whether the left childen is larger or smaller than the right one. Therefore, the min-heap property cannot be used to print out the keys in sorted order in $O(n)$ time. We fail to find method of confirm which is the smallest.

6 Binary search tree

From (a) we have the following figure ...,therefore, (a) is the right sequences. Similary, for (b),we have the following binary tree that meet the standand of binary search tree. It is shown in figure ....

For (c), we cannot find the right type of binary tree, when we take the left child from the number 911 node, we cannot construct the 912, because 912 is larger than 911, it cannot be considered as the left children.

For (d), we can esbablish the following figure ..., which can be right to meet the conditions.

For (e), we cannot build the tree, we can take the right node at 347, after that, we will add the 299 node into the tree, where will produce collision, therefore, it is wrong.

From what have been discussed above, we can find (a),(b),(d)is right and (c),(e) is wrong.

7 Euler's Formula

As we can see, the solid has genus zero, and each face is either a square or a pentagon, if we know the number of squares is S, and the number of pentagons is P,3 faces meet at each vertex.Every edge will be common for every face. For squares, we have vertex and edges: V1=4S/3,E1=4S/2. For pentagons, we have vertex:V2=5P/3,E2=5P/2; the total vertex is V=V1+V2; E=E1+E2; The total face for the solid is F=S+P; According to the Euler's Formula: V-E+F=2. We substitute the above expression into this formula, and we can get 2S+P=12.Therefore, it is proofed.

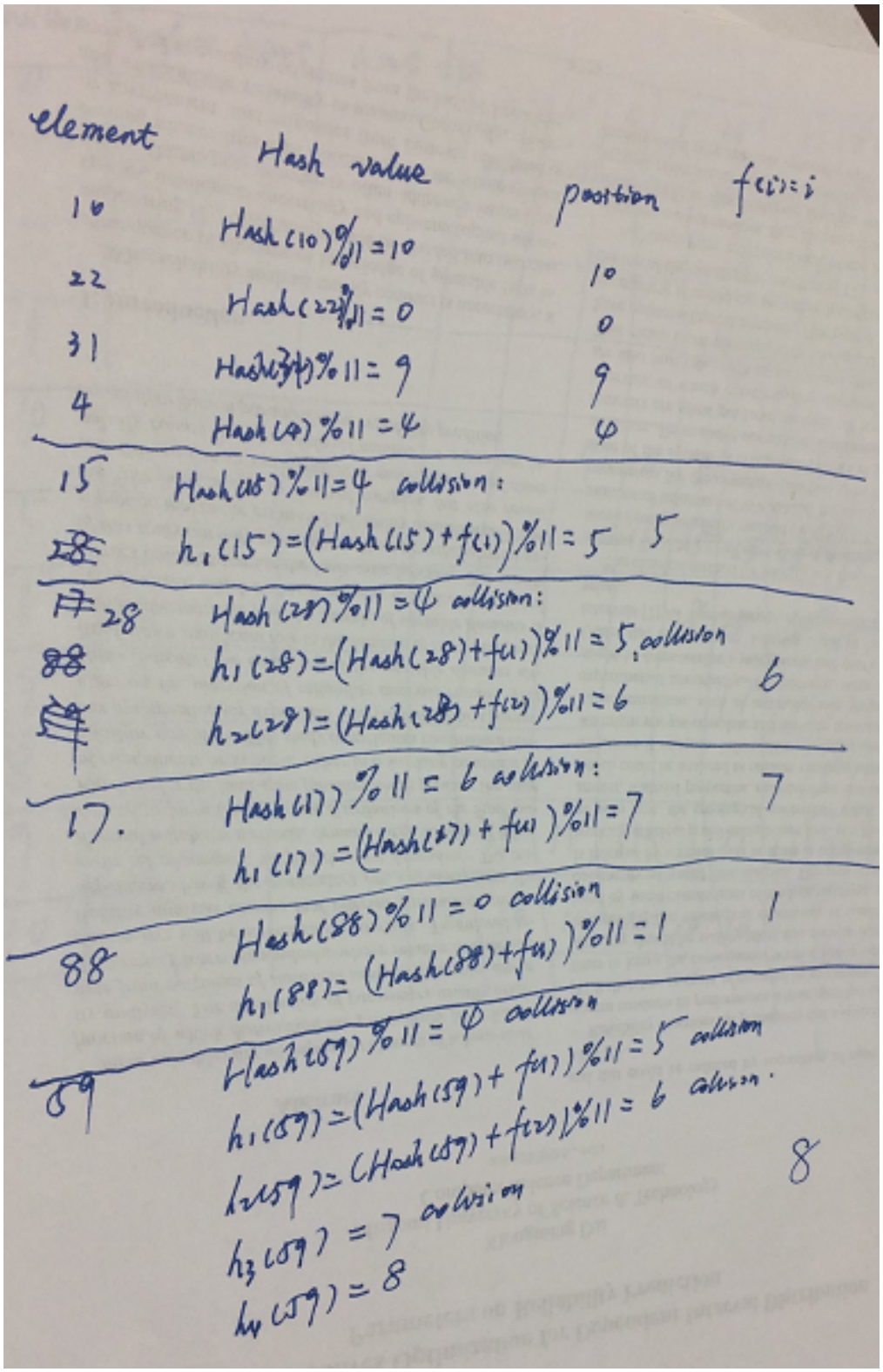


Figure 2: linear probing 1

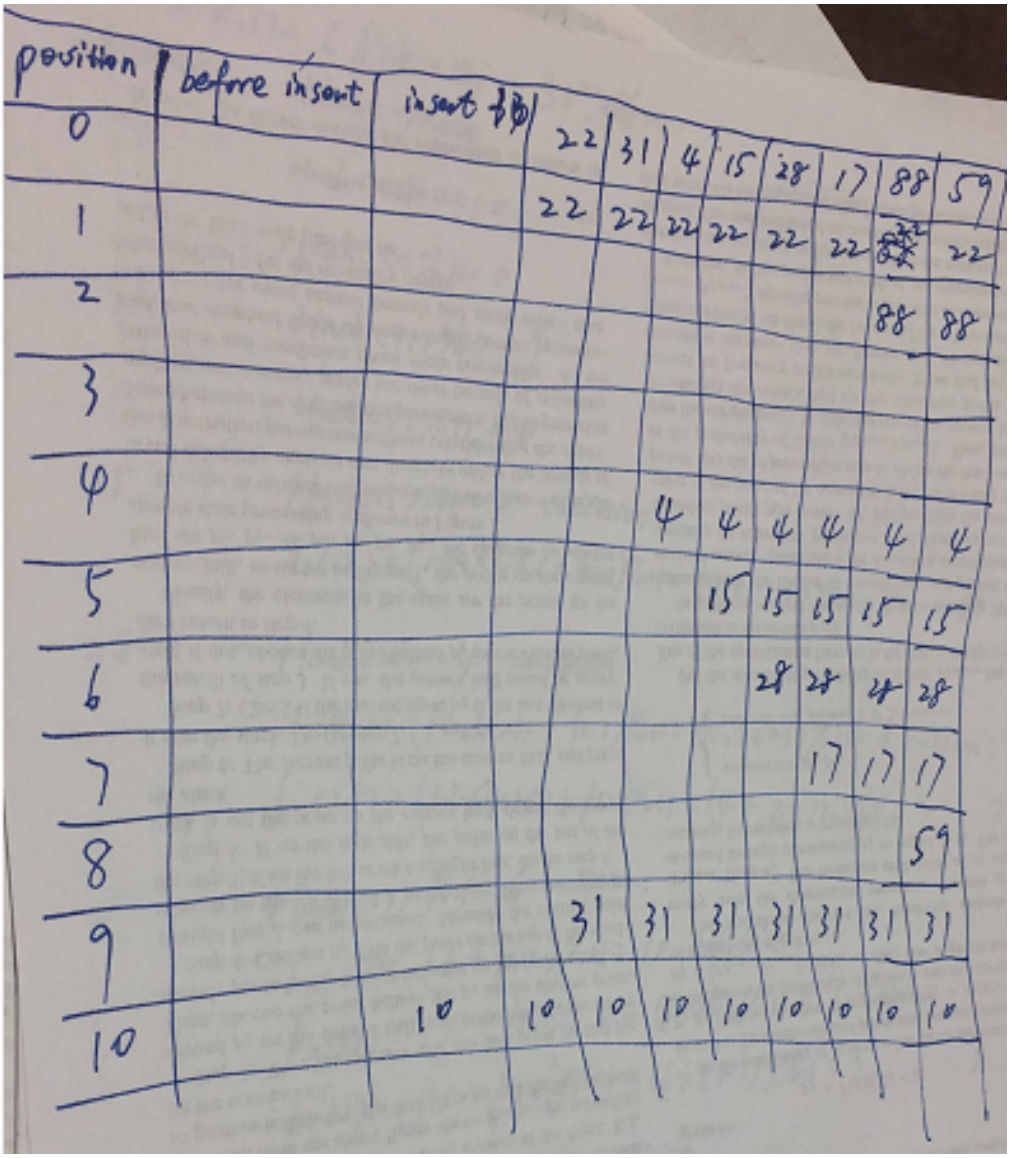


Figure 3: linear probing 2

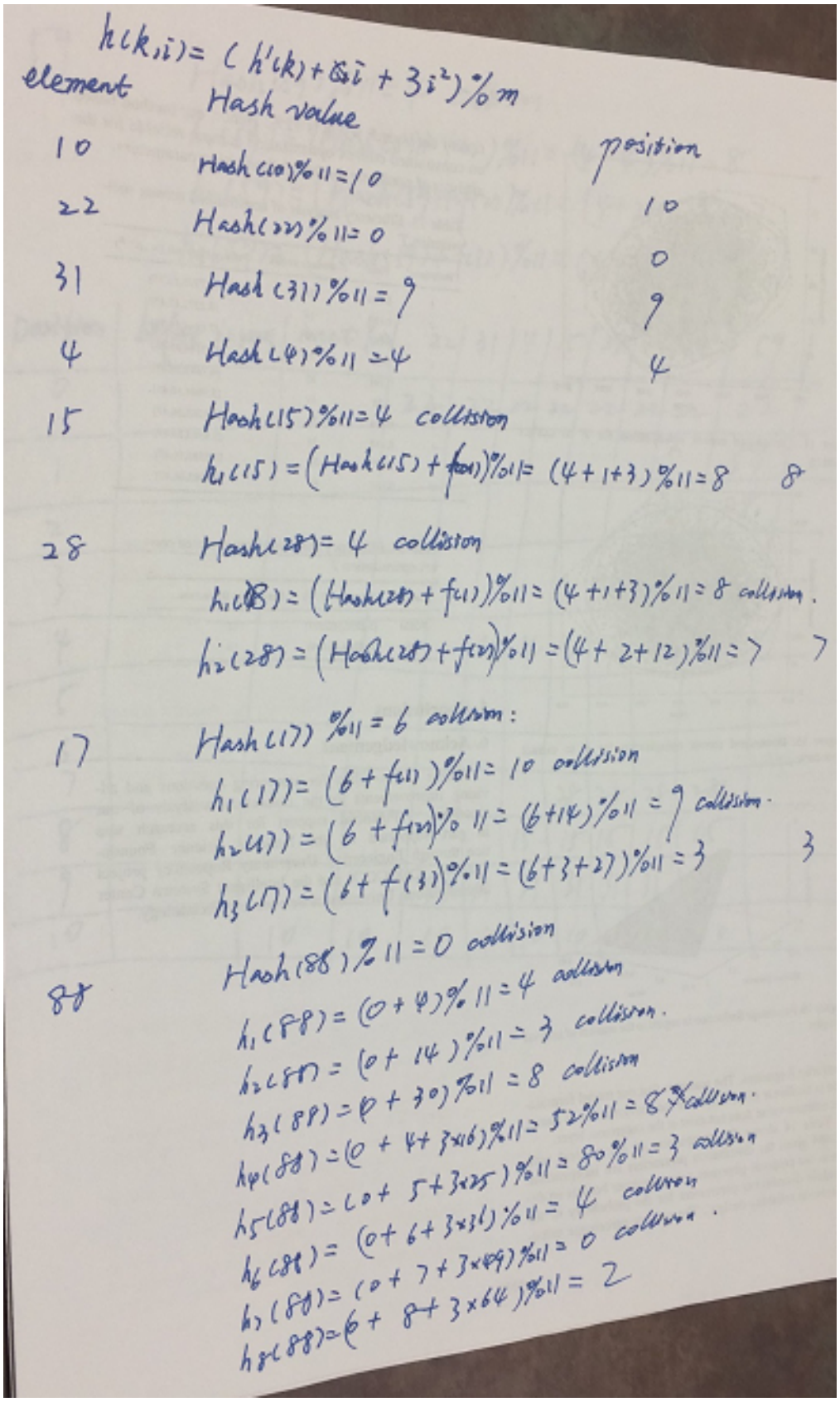


Figure 4: quadratic probing 1

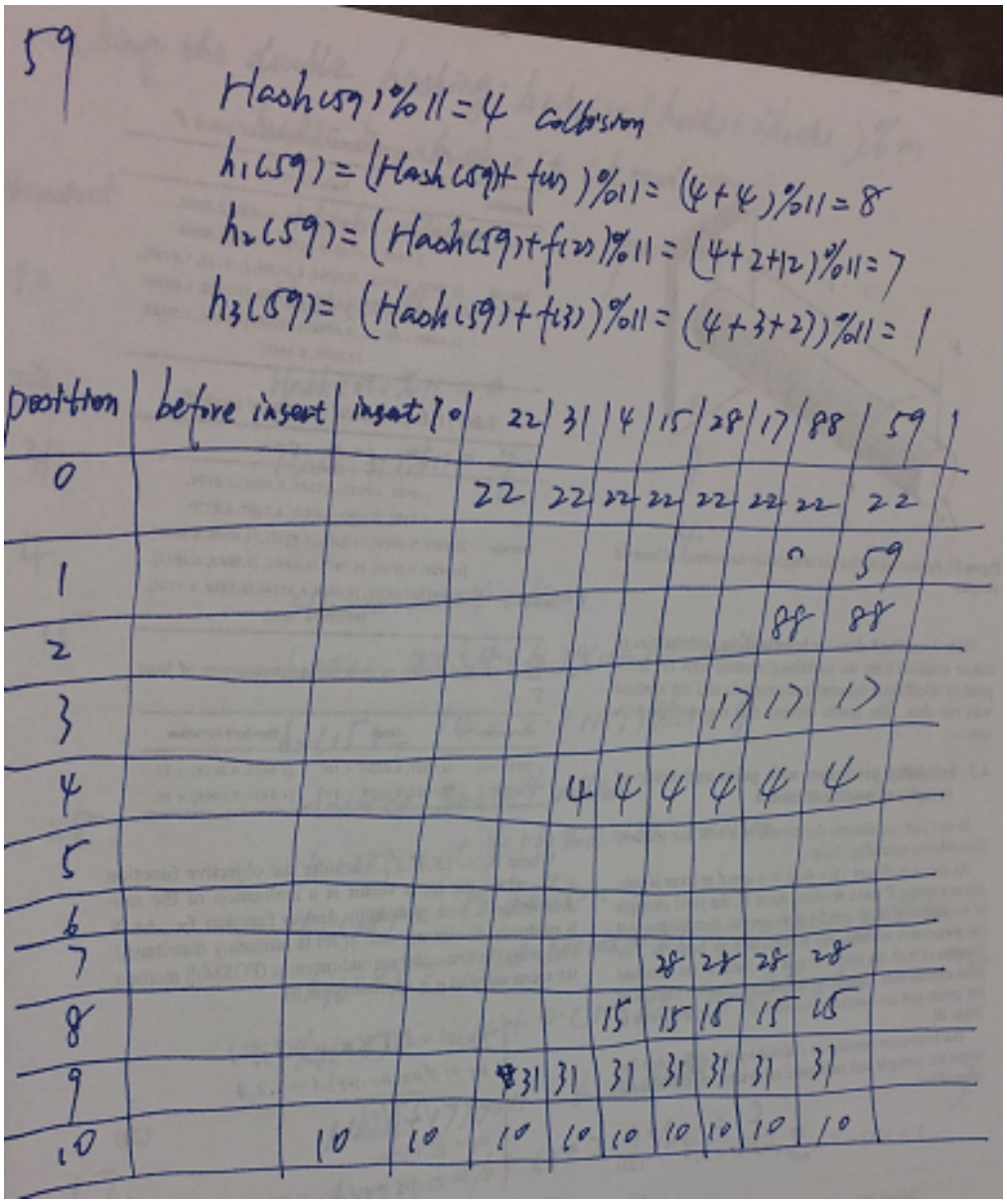


Figure 5: quadratic probing 2

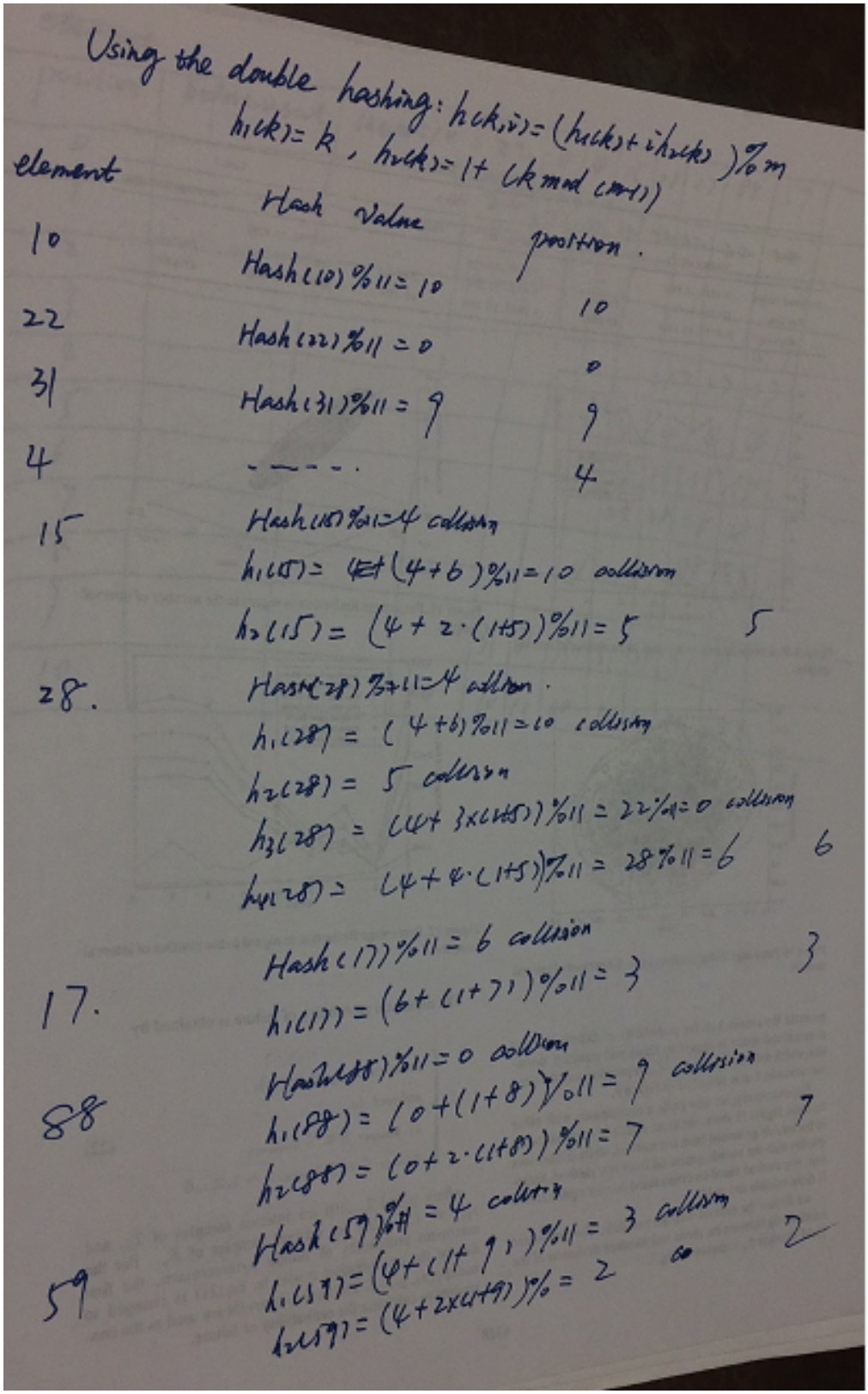


Figure 6: double hashing 1

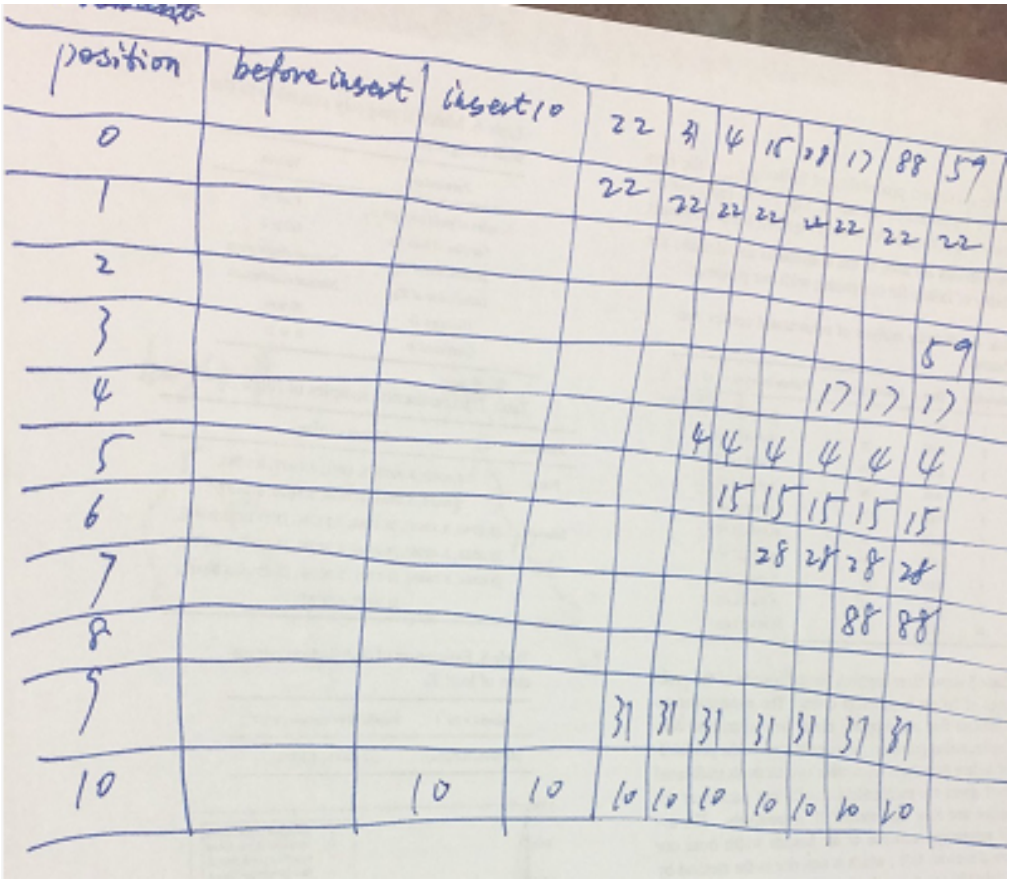


Figure 7: double hashing 2

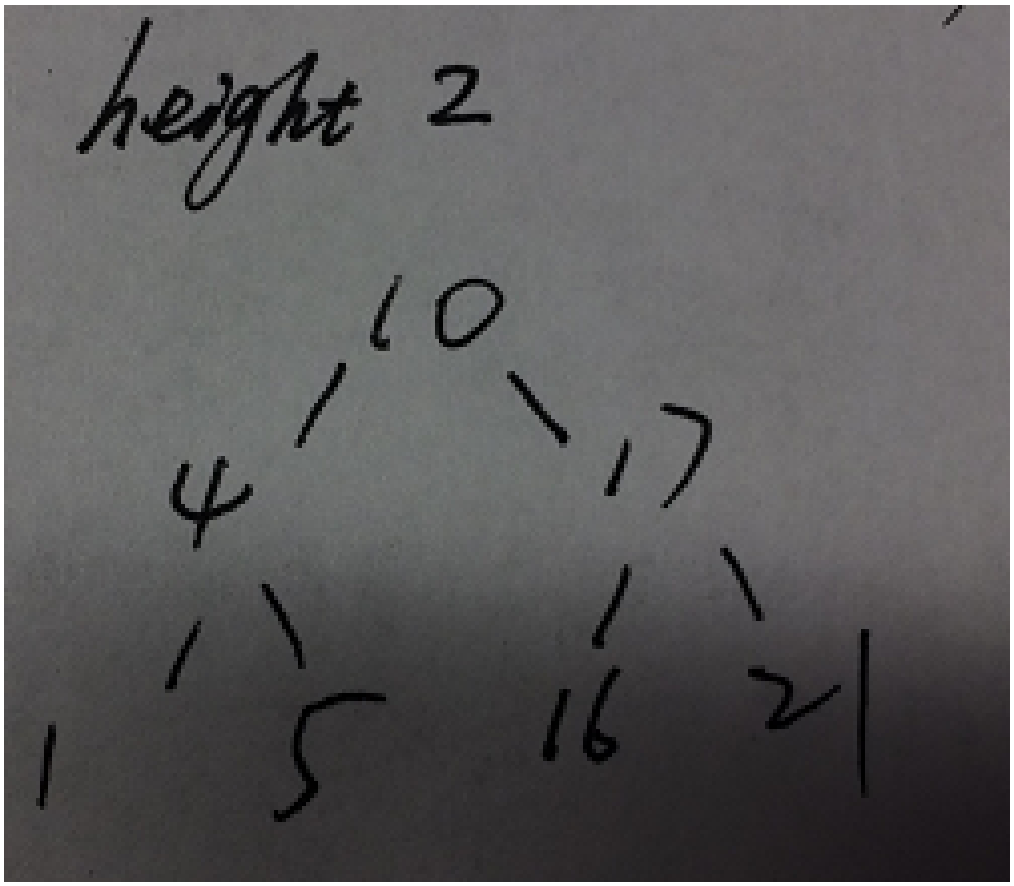


Figure 8: height 2

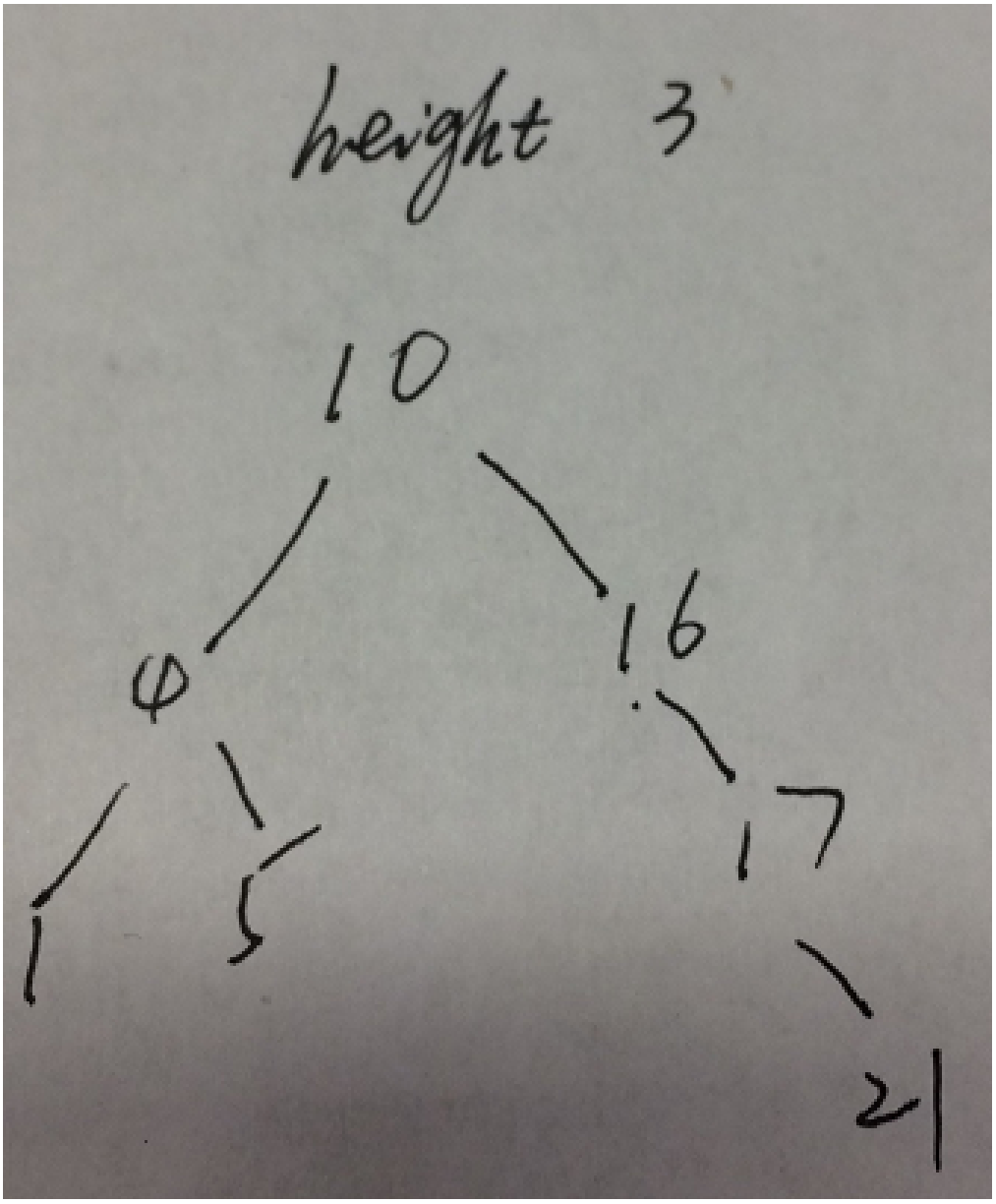


Figure 9: height 3

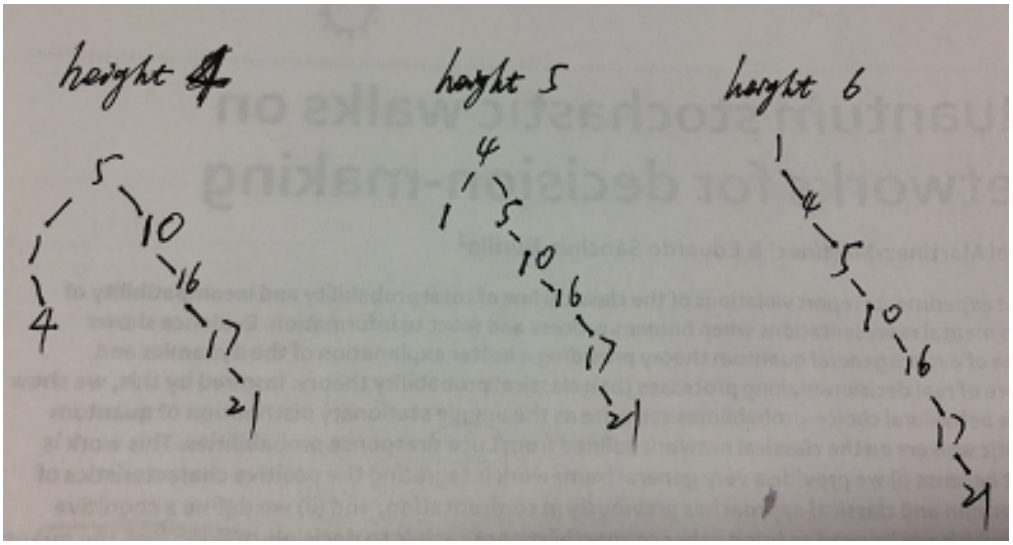


Figure 10: height 4,5,6

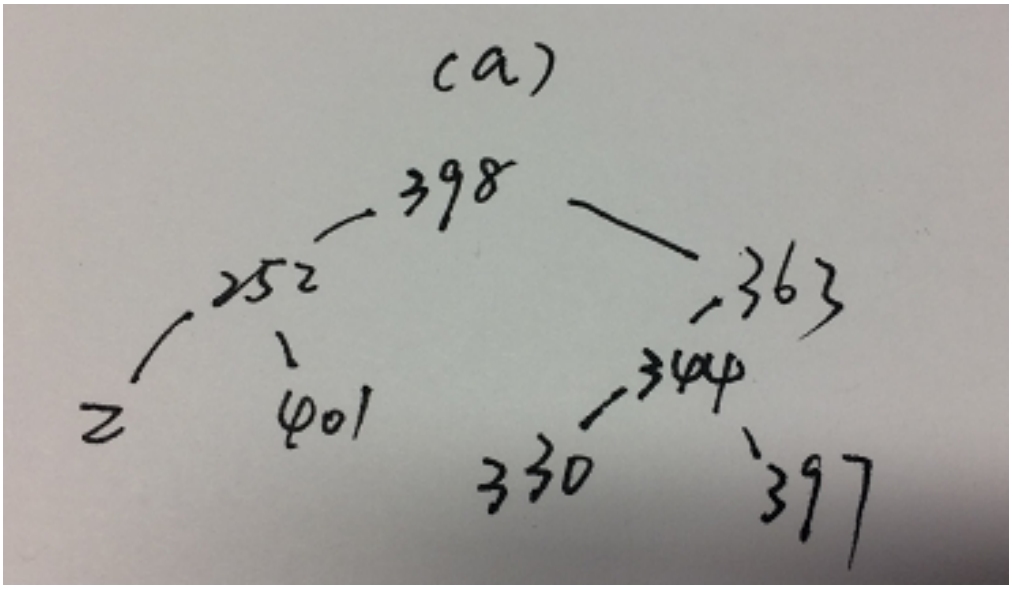


Figure 11: lists

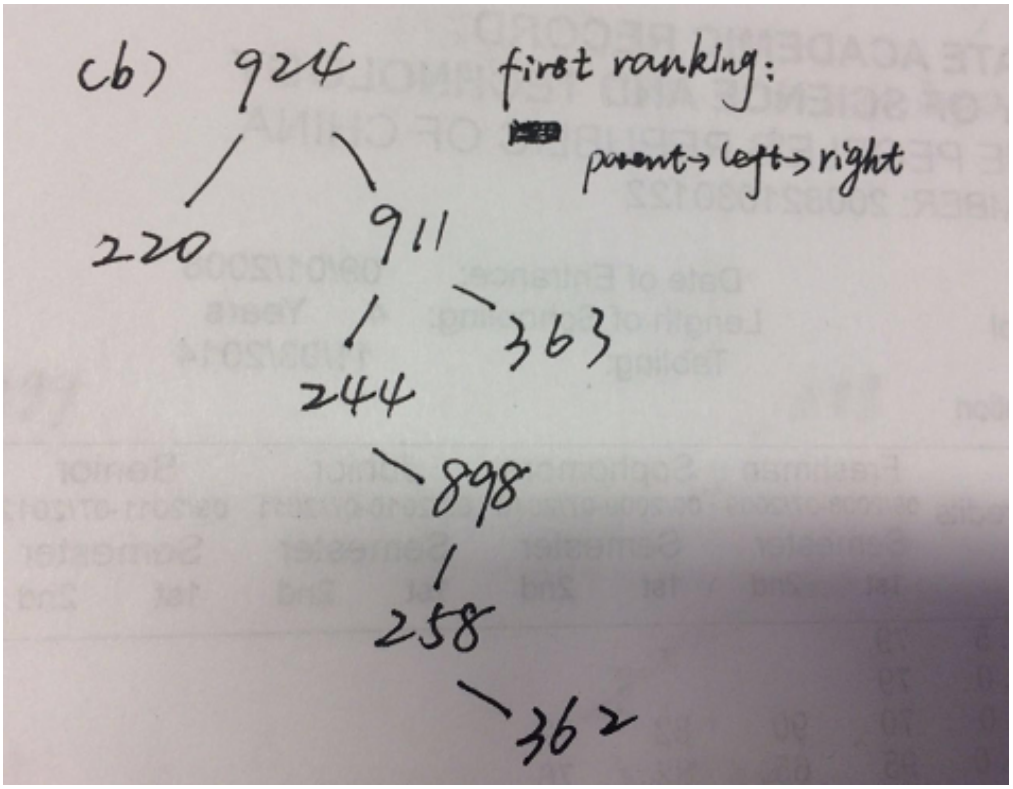


Figure 12: lists

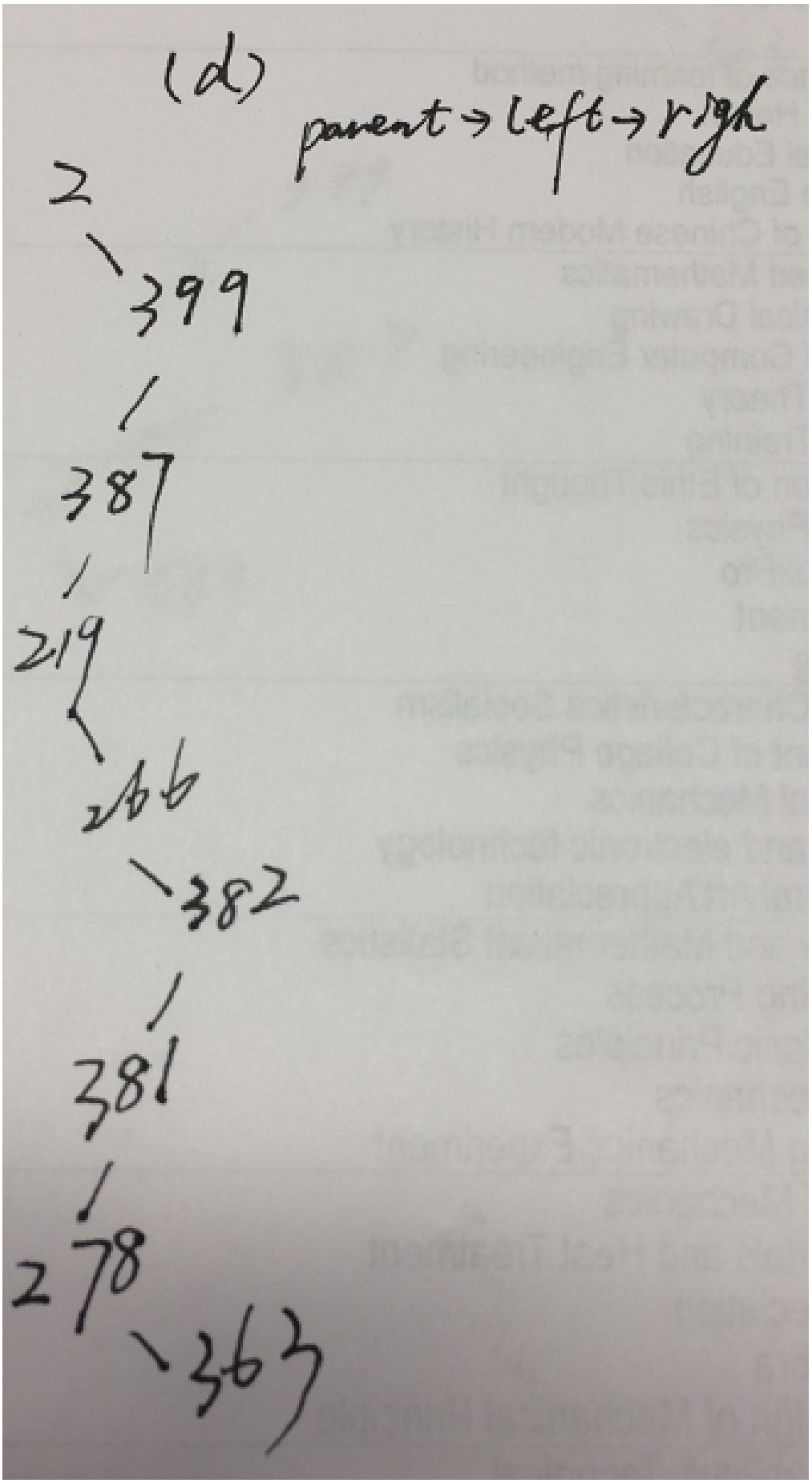


Figure 13: lists