Exam 2

Graded

Student

Adam Fenjiro

Total Points

73.15 / 100 pts

Question 1

Question 1 — **27** / 30 pts

1.1 (no title) — **1** / 1 pt

✔ **– 0 pts** Correct

– 1 pt Incorrect

1.2 (no title) — **1** / 1 pt

✔ **– 0 pts** Correct

– 1 pt Incorrect

1.3 (no title) — **1** / 1 pt

✔ **– 0 pts** Correct

– 1 pt Incorrect

1.4 (no title) — **1** / 1 pt

✔ **– 0 pts** Correct

– 1 pt Incorrect

1.5 (no title) — **1** / 1 pt

✔ **– 0 pts** Correct

– 1 pt Incorrect

1.6 (no title) — **1** / 1 pt

✔ **– 0 pts** Correct

– 1 pt Incorrect, correct answer was B

1.7 (no title) — **2** / 2 pts

✔ **– 0 pts** Correct

– 1 pt Incorrect, should be A: In order

1.8 (no title) — **2** / 2 pts

✔ **– 0 pts** Correct

– 1 pt Partial Credit

– 2 pts Incorrect, Should be A,C

1.9 (no title) — **2** / 2 pts

✔ **– 0 pts** Correct

– 2 pts Incorrect, should only be b

1.10 **(no title)** **2** / 2 pts

✔ **– 0 pts** Correct

– 1 pt Partial

– 2 pts Incorrect, Should be C,D

1.11 **(no title)** **2** / 2 pts

✔ **– 0 pts** Correct

– 2 pts Incorrect

– 1 pt Partial Credit

1.12 **(no title)** **2** / 2 pts

✔ **– 0 pts** Correct

– 2 pts Incorrect

– 0.5 pts 3 correct

– 1 pt 2 correct

– 1.5 pts 1 correct

1.13 **(no title)** **3** / 3 pts

✔ **– 0 pts** Correct

– 1 pt Missed a correct answer, or selected additional wrong answer

– 2 pts Missed 2 correct answers

– 3 pts Incorrect

1.14 **(no title)** **3** / 3 pts

✔ **– 0 pts** Correct

– 3 pts Incorrect

1.15 **(no title)** **0** / 3 pts

– 0 pts Correct

✔ **– 3 pts** Incorrect

1.16 **(no title)** **3** / 3 pts

✔ **– 0 pts** Correct

– 3 pts Incorrect

Question 2

(no title) **3** / 10 pts

2.1 **(no title)** **1** / 1 pt

- ✔ **– 0 pts** Correct
- **– 1 pt** Incorrect
- **– 0.5 pts** Off by 1 level

2.2 **(no title)** **1** / 1 pt

- ✔ **– 0 pts** Correct
- **– 1 pt** Incorrect, Should have been 15
- **– 0.5 pts** Off by 1 level. Should have been 15

2.3 **(no title)** **0** / 2 pts

- **– 0 pts** Correct
- ✔ **– 2 pts** Incorrect
- **– 1 pt** partial credit

2.4 **(no title)** **1** / 6 pts

- **– 0 pts** Correct
- **– 0.5 pts** Typo level bug
- **– 1 pt** parent() Is incorrect
- **– 1 pt** Issue with boundary condition on upheap
- **– 2 pts** Minor bug
- **– 4 pts** Major bug or multiple bugs with upheap
- ✔ **– 5 pts** Upheap is incorrect
- **– 6 pts** Incorrect / Blank

Question 3

(no title) **8** / 10 pts

3.1 **(no title)** **4** / 6 pts

- **– 0 pts** Correct
- **– 1 pt** Minor Error in Tree
- ✔ **– 2 pts** wrong tree, need to combine lowest frequencies first
- **– 2 pts** wrong tree, only leaf nodes contain letters
- **– 2 pts** 3.1.2 should be 91
- **– 2 pts** 3.1.3 should be 27
- **– 6 pts** Incorrect

💬 Following the tree algorithm, l needs to be matched with the p/n tree.

3.2 **(no title)** **4** / 4 pts

- ✔ **– 0 pts** Correct
- **– 1 pt** Minor errors in part 3.2.1
- **– 2 pts** No answer or incorrect tree in part 3.2.1
- **– 2 pts** No answer or incorrect answer in part 3.2.2

Question 4

(no title) **9.5** / 10 pts

4.1 **(no title)** **4** / 4 pts

- ✔ **– 0 pts** Correct
- **– 4 pts** Incorrect

4.2 **(no title)** **3** / 3 pts

- ✔ **– 0 pts** Correct
- **– 1 pt** Two Letters off
- **– 2 pts** 4 Letters off
- **– 3 pts** Incorrect, Should have been blank,B,D,A,C,E,F,G

4.3 **(no title)** **2.5** / 3 pts

- **– 0 pts** Correct
- ✔ **– 0.5 pts** Small error
- **– 1.5 pts** Partial credit
- **– 3 pts** Incorrect/no answer

💬 E loops to 0 as part of the modulus operator

Question 5

(no title) **7** / 10 pts

5.1 **(no title)** **1** / 1 pt

✔ **– 0 pts** Correct

– 1 pt incorrect

5.2 **(no title)** **0** / 2 pts

– 0 pts Correct

✔ **– 2 pts** Incorrect

– 1 pt Partial credit

For each node in a BST, all nodes in the right subtree must be greater than that node, and all nodes in the left subtree must be less than that node.

5.3 **(no title)** **4** / 4 pts

✔ **– 0 pts** Correct

– 1 pt Final tree has minor errors

– 1 pt Tree after removing 20 has minor errors

– 1 pt Inconsistent replacement policy for nodes

– 2 pts Did not remove 40 from tree

– 4 pts Incorrect

5.4 **(no title)** **2** / 3 pts

– 0 pts Correct

– 1 pt Slight inaccuracy with sentinel nodes

✔ **– 1 pt** Height is off-by-one

– 2 pts Tree is either not a bst or does not fit requirements

– 1 pt Tree contains invalid data

– 3 pts Incorrect/no answer

Question 6

(no title) **6.4** / 10 pts

6.1 **(no title)** **3.5** / 5 pts

- **– 0 pts** Correct
- **– 1 pt** minor error
- **– 1 pt** First bound should be p
- ✔ **– 1.5 pts** q-1 and q+1 are the inner bounds since we need to remove the pivot in recursive calls to ensure progress.
- **– 1 pt** Fourth bound should be r
- **– 1 pt** Should provide the same array to both calls
- **– 5 pts** Incorrect
- **– 3 pts** Args do not use q properly.

6.2 **(no title)** **1.4** / 2 pts

- **– 0 pts** Correct
- **– 0.6 pts** Missed 1 pivot, should be 4,5,9
- ✔ **– 0.6 pts** Added an extra pivot, should be 4,5,9
- **– 1.3 pts** Missed 2 pivots, should be 4,5,9
- **– 1.3 pts** Added two extra pivots, should be 4,5,9
- **– 2 pts** Incorrect, should be 4,5,9

6.3 **(no title)** **1.5** / 3 pts

- **– 0 pts** Correct
- ✔ **– 1.5 pts** Half Correct, should be 6,3,5,1,4,2,7,8,9,10
- **– 3 pts** Incorrect, Should be 6,3,5,1,4,2,7,8,9,10

Question 7

(no title) **7.25** / 10 pts

7.1 **(no title)** **1** / 1 pt

✔ **– 0 pts** Correct

– 0.5 pts Partial Credit

– 1 pt Incorrect

7.2 **(no title)** **0.5** / 1 pt

– 0 pts Correct

✔ **– 0.5 pts** Partial Credit

– 1 pt Incorrect

The range is h-l, so if you simply divide h in half, you do not account for the offset of l units included in h. mid = (l+h)/2

7.3 **(no title)** **0.75** / 1 pt

– 0 pts Correct

✔ **– 1 pt** Incorrect

+ 0.75 pts This will cause the loop to continue iterating forever since the loop will run with l=h. It's easier to just return here with "return mid"

7.4 **(no title)** **0.5** / 1 pt

– 0 pts Correct

✔ **– 0.5 pts** Partial Credit

– 1 pt Incorrect

This is the case in which we should enter the upper half, not the bottom half

l=mid+1

7.5 **(no title)** **0.5** / 1 pt

– 0 pts Correct

✔ **– 0.5 pts** Partial Credit

– 1 pt Incorrect

This should be h=mid+1

7.6 **(no title)** **1** / 1 pt

✔ **– 0 pts** Correct

– 0.5 pts Partial Credit

– 1 pt Incorrect

Careful, since mid can be off-by-one, l is a more proper return since at this point we know that k does not exist in this map.

7.7 **(no title)** **1** / 2 pts

– 0 pts Correct

– 0.5 pts Minor Mistake at One Step

✔ **– 1 pt** Terminates early or late

– 1 pt Wrong Key

– 2 pts Incorrect

7.8 **(no title)** **2** / 2 pts

✔ **– 0 pts** Correct

– 0.5 pts Minor error

– 1 pt Returned a not-found code instead of 13

– 2 pts Incorrect

Question 8

(no title) **5** / 10 pts

– 0 pts Function is correct and takes O(n+m) time

– 0.5 pts Logic is correct in the case of no ties and takes O(n+m) time, but only returns one winner.

– 0.5 pts Function is correct and takes O(n+m) time, but has a minor error with assembling the final solution

– 2 pts Function reasonably solves the provided problem, and while it does not take O(n+m) time, a correct or mostly correct time complexity is provided

– 3 pts Function has minor issues, but is close to the O(n+m) solution

– 4 pts Function is correct, but does not take O(n+m) time and the provided time complexity is wrong.

– 4.5 pts Function has minor issues and does not take O(n+m) time, but has a correct time complexity provded.

✔ **– 5 pts** Function has minor issues and a wrong time complexity.

– 7 pts Function has major issues.

– 10 pts Blank or Incorrect

O(n^2 + m^2), and does not return the proper output

Data Structures, Spring 2023

Total: 90 + 10 extra points

Please note there are 10 extra points, but no designated questions for extra points. This is a just way to booster your grade. Please take the time to work on the problems and get the most points.

Please write your name and user name clearly. Account name/User name is the part before @ in your email address. It is not your M number. For example, my email is ruihong@mtu.edu and my username is ***ruihong***

Your Name: Adam FENJIRO

Account Name/User Name: afenjiro

1 (30 p) Fill the blanks with letter(s) of the multiple choices.

1) (1 point) What is the best definition of collision in a hash table implementation of MAP? C
   A. Two key-value pairs that have equal keys but different values.
   B. Two key-value pairs that have different keys and hash to different indices.
   C. Two key-value pairs that have different keys but hash to the same index.
   D. Two key-value pairs that have equal keys but hash to different indices

2) (1 point) What is the expected time complexity of get(k) for hash table? C
   A. O(n) B. O(log n) C. O(1)

3) (1 point) What is the worst-case time complexity of get(k) in a binary search tree? A
   A. O(n) B. O(log n) C. O(1)

4) (1 point) What is the worst-case time complexity of deleting an entry in a sorted array? A
   A. O(n) B. O(log n) C. O(1)

5) (1 point) What is the worst-case time complexity of deleting an entry in a binary search tree? A
   A. O(n) B. O(log n) C. O(1)

6) (1 point) The lower bound running time of any comparison-based algorithm for sorting an n-element array is B
   A. $\Omega(n^2)$ B. $\Omega(n \log n)$ C. $\Omega(1)$

7) (2 point) Which traversal will print the key from min to max in a binary search tree? A
   A. in order B. post order C. pre order

8) (2 points) Which of the following (s) are the causes for ineffective hash table implementation for MAP ADT? Choose all the correct ones: A, C
   A. Bad hash function
   B. Too many duplicate keys
   C. Hash table size is too small

9) (2 points) Which sorting algorithm(s) has a lower bound of $\Omega(n)$? Choose all correct ones: B
   A. selection sort
   B. insertion sort
   C. merge sort
   D. quick sort

10) (2 points) Which sorting algorithms use divide and conquer technique? Choose all correct ones: C, D
   A. Selection Sort
   B. Insertion Sort
   C. Merge Sort
   D. Quick Sort

11) (2 points) What data structures are the **reasonable** choices to implement MAP<K,V> ADT when K is **NOT** comparable for the order? Please select all correct ones. A, C

A. Array List
B. Sorted Array List
C. Hash Table
D. Binary Search Tree

12) (2 points) Which statement(s) are true about prefix code? Choose all correct ones: A, B, C, D

A. no code word is a prefix of other code word
B. Huffman code is prefix code
C. prefix code is variable length
D. prefix code can be decoded unambiguously without using special markers between code word

13) (3 points) Suppose that we have numbers between 1 and 100 in a binary search tree. We want to search for 55. Which of the following sequence(s) could be the sequence of nodes examined? Please select all the correct ones.

A, B, D

Hint: Remember in binary search tree:

1. All keys in the left subtree has to be smaller than the key of root.
2. All keys in the right subtree has to be greater than the key of the root.
3. The above condition has to be met recursively for all subtrees as well.

A. 10, 75, 64, 43, 60
B. 90, 12, 68, 34, 62
C. 9, 85, 47, 68, 43
D. 79, 14, 72, 56, 16

14) (3 points) For heap PQ sort, the first step is to make a minimum heap out of the array. Given array [5,4,3,2,1], what the array would be after the first step? C

A. 1, 3, 2, 5, 4
B. 1, 2, 3, 4, 5
C. 1, 2, 4, 5, 3
D. 1, 2, 5, 4, 3

15) (3 points) After inserting (5, A), (7, D), (6, C), (2, E), (3, F), (1, B) one by one to a PQ implemented by a minimum heap, what is the data in the array of the heap? C

A. (5, A), (1, B), (6, C), (7, D), (2, E), (3, F)
B. (1, B), (2, E), (3, F), (5, A), (6, C), (7, D)
C. (1, B), (2, E), (3, F), (7, D), (5, A), (6, C)
D. (1, B), (3, F), (2, E), (7, D), (5, A), (6, C)

16) (3 points) What does the array of a heap [1, 3, 2, 6, 5, 4 ,8 ,7] become after removing the minimum? B

A. 2, 3, 4, 6 ,7, 8, 5
B. 2, 3, 4, 6, 5, 7, 8
C. 2, 4, 3, 6, 5, 7, 8
D. 2, 3, 4, 5, 6, 7, 8

2 (10 p) Heap and its array implementation

2.1. (1p) What is the minimum number of nodes in a heap of height 3? 8

2.2. (1p) What is the maximum number of nodes in a heap of height 3? 15

2.3. (2p) Given a heap with 100 nodes, what is the height of the heap? $\log_2(111) - 1$

2.4. (1p) Complete the functions: parent(i)

```
Alg parent(i)
Output: return the parent index of node i
    return floor(i-1 / 2);
```

2.5. (5p) Complete the following and upheap(i)

- Assume we use **`Arraylist<Entry<A,V>> data`** as the storage for the heap
- You may use all the methods of arraylist,
- You may also use <,>,= to compare the keys.
- You MUST implement it as recursive function. Don't use for-loop, while-loop.
- You may use `swap(A,i,j)` to swap the entry at index i and j

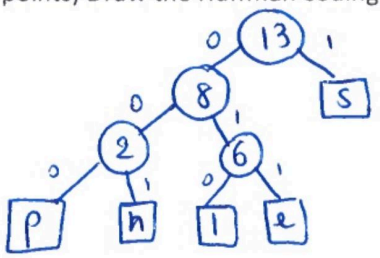
```
Alg UpHeap(i)
      Input: All nodes of A satisfy the min heap order except index i, that is,
        A.get(i)'s key value might be smaller than its parent's key value.
      Output: the heap-order is restored for A
```

3. (10 points) Encode, Decode and Text compression

3.1. For the word: **sleeplessness**, the frequency of each character has been counted as in the following table:

letter	s	l	e	p	n
frequency	5	2	4	1	1

3.1.1. (2 points) Draw the Huffman coding tree:



S = 1
e = 011
l = 010
n = 001
p = 000

3.1.2. (2 point) If we use ASCII coding system where each character will be encoded using 7 bits, how many digits will be needed to represent the word **sleeplessness**?

7 x 13 = 91 digits

3.1.3. (2 point) How many bits of 1's and 0's in the binary code representation of the input using Huffman coding?

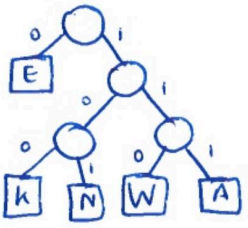
(1 x 5) + (4 x 3) + (2 x 3) + (1 x 3) + (1 x 3)

= 29 bits

3.2. (4 points) Encode and Decode

3.2.1. Draw the binary coding tree for the coding system which contains 5 letters

E: 0 K: 100 N: 101 W: 110 A: 111



3.2.2. Decode 1 0 0 0 0 1 1 0 0 0 1 0 1 1 1 1 1 1 0 using the coding system in 3.2.1

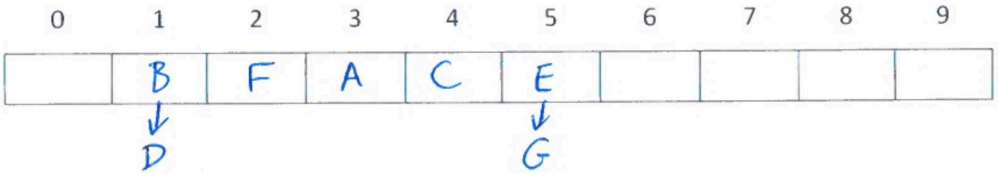
K e e w e e n a w

4. (10 points) Hash

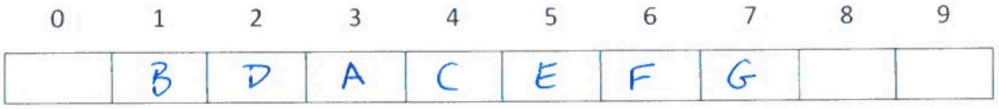
Insert the following keys A, B, C, D, E, F, G one by one to an initially empty hash table of size **10**, using the following hash value.

K	A	B	C	D	E	F	G
h(k)	3	1	4	1	5	2	5

4.1. (4 points) Use the **separate chaining** method to handle collision. Draw the hash table with all the keys in it.

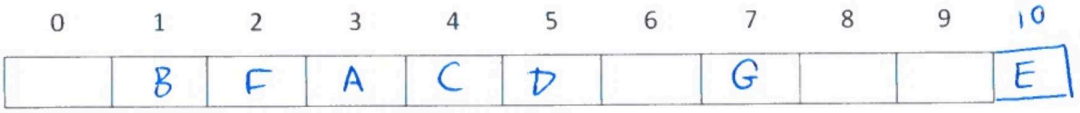


4.2. (3 points) Use the **linear probing** to handle collision. Draw the hash table with all the keys in it.



4.3. (3 points) Use the **double hashing probing** to handle collision. Draw the hash table with all the keys in it. The second hash value h'(k) for each key has been precalculated for you:

K	A	B	C	D	E	F	G
H'(k)	1	2	3	4	5	6	2

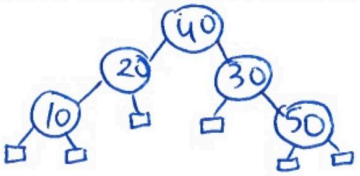


5. (10 points) Binary Search Tree

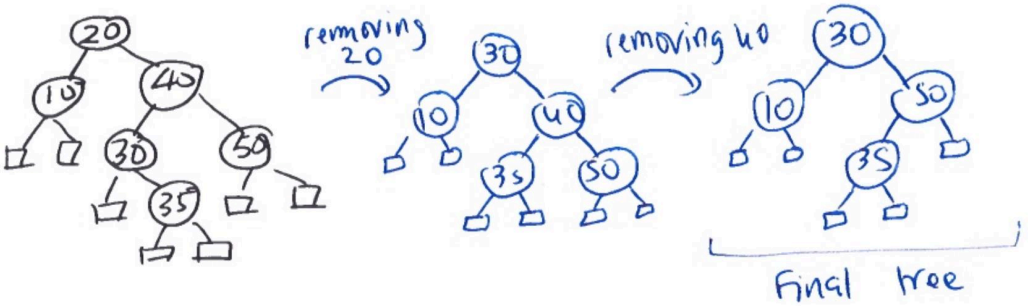
5.1. (1 points) We use the sentinel nodes when we implement the binary search tree. If there are ***n*** data node, how many sentinel nodes?

nSentinel = n + 1 ✓

5.2. (2 points) Put the following data into an empty binary search tree: 40, 20, 10, 30, 50. Draw the final tree. Please **include the external sentinel nodes.**

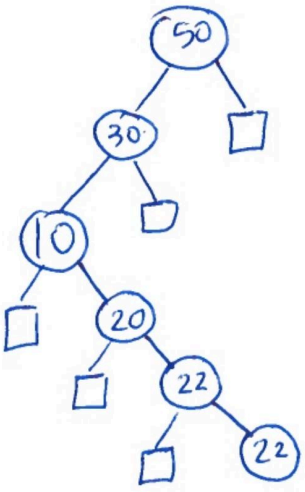


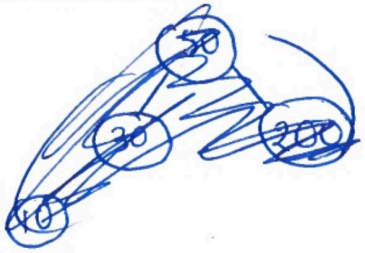
5.3. (4 points) Remove 20 first, then remove 40 from the binary search tree below and mark the final tree clearly. Use **successor** to replace the entry in the tree node if needed.



5.4. (3 points) Draw a binary search tree with the following requirement

5.4.1. The external nodes are used as sentinel nodes
5.4.2. The height of the tree is 5 with 5 internal nodes
5.4.3. The tree stores only 5 integer keys.
5.4.4. Show the keys with some valid data





6. (10 points) Sorting Algorithm

6.1. (5 points) Finish the following pseudo code for Quick Sort.

```
Algorithm QuickSort(A,p,r)
   Input: Array A, p and r specify the subarray A[p..r]
   Output: A[p..r] is sorted

   if p < r then
      q ← Partition(A,p,r)

      // TODO: Recursively call quicksort to sort the two parts after partition
```

QuickSort(a, p, q)
QuickSort(a, q+1, r)

6.2. (2 points) Here is an array which has just been partitioned by the first step of quicksort:

3	0	2	4	5	8	6	9

Which of these elements could be the pivot that was used for the partition? There may be more than one possibility! List all of them.

3, 4, 5, 9

6.3. (3 points) Given the array

9	3	5	1	4	10	2	8	6	7

Show the array after partition using the last element 7 as pivot. You need to use the algorithm that we studied in class.

6	3	5	1	4	2	7	10	8	9

7. (10 points) Binary Search

7.1. (6 points) Finish the following pseudo code for binary search in sorted array. Please note that we are using whole loop, not recursive calls.

```
Algorithm BinarySearch(A, n, k,)
   Input: Sorted Array A of n items and search key k
   Output: return the index where k exist in the array.
           if k does not exist, return the index where k should exist

   l ← 0
   h ← n-1

   // while the search range is not empty

   while  l < h  do
          //set mid to be the middle index of the search range
          mid = h/2 +1
          If  A[mid] = k then
              h = l
          else if A[mid] < k then
              h = mid
          else
              l = mid

   return A[mid]
```

7.2. (4 points) Given a sorted array A of 15 numbers below

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	3	4	10	11	20	23	31	34	42	55	56	61	68	88

Call the above function binarySearch(A, 15, 67)

What keys will be compared with the search key 67? List them in the order when it will be compared with 67: 31, 56, 68

What does the above function call return? 13

8. (10 points) Who is the most improved athlete of the year?

There are n athletes. Each athlete is assigned with a unique number: 1,2,3,...n. There are total of m votes. The votes are recorded in the array **vote**. The function winner() returns the highest number of votes that a single athlete received, followed by the winner(s) (the athletes who received the highest number of votes). To get full points, the worst case time complexity should be O(n+m). If your algorithm works correctly and you analyzed the time complexity correctly but not O(n+m), you will receive 80%.

m = size of array
n = range
return highest count, n-count

```
Example 1:
Input:  n=5, m=10, votes = [1, 2, 2, 2, 1, 2, 1, 1, 1, 1]
Output: 6, 1
      Return doubly linked list of two elements: 6, 1
      The first number means the highest number of votes is 6.
      The second athlete No. 1 received the highest votes.


Example 2:
Input:  n=10, m=8, votes = [1, 2, 2, 5, 2, 1, 4, 1],
Output: 3, 1, 2
      Return doubly linked list of three elements: 3, 1, 2
      The first number means the highest number of votes is 3.
      The second and third number represents both athlete No. 1 and athlete No.2
received the highest votes.
```

Please write your answer on the next page

- The function header is given
- The first line and last line of pseudo code are given
- Please write comment for your pseudo code
- Please use consistent indentation

- Please write inside the block
- Please write as neat as possible

8.1. (6 points) Write the pseudo code for the function winner(). You may use any data structures and algorithms directly that we studied in class. **Hint**: Use the idea of bucket sorting algorithm.

```
Alg winner(n, m, votes)
    Input: n athletes, m votes, Array votes,
    Output: Returns a doubly linked list. The first number is the highest vote, followed by the numbers of the
athletes who received the highest number of votes.

  output <- new DoublyLinkedList();

  // vmap that contains the vote a and the number of times b.
  vmap<a, b> = new map<int, int>();
  int high = 0;
  for (int i=0; i<m; i++){                    // if there's no vote, we add it
      if (vmap.getKey(votes[i]) == null){     and set it's count to 1.
          vmap.put(votes[i], 1);              // if there is a vote, we +1 the
      } else { vmap.put(votes[i], vmap.getKey(votes[i])+1);}   count
  }
  for (int i=0; i<n; i++){                    // we compare the counts and
      if (vmap.getKey(i) > high){             assign the highest count = high
          high = vmap.getKey(i);
      }
  }
  return high + ", " + vmap.getKey(high);

  return output
```

8.2. (2p) What is the worst case time complexity of your function winner() in terms of **n and m**?

$O(n^2)$

8.3. (2 points) Is the time complexity O(n+m)?

No