COMP3121/3821/9101/9801 18s1 | Assignment 1 (UNSW)

Student name: SHIQING ZHANG zid: z5119751

1.

(a)

First, we need to sort array’s element by using merge sort in O(nlogn).

Second, by using while loop:

head is the point which starts from 0, increasing one by one once

tail is the point which starts from the number of array elements, decreasing one by one once

when head is over than or equal to tail, while loop break

otherwise goes into three situations’ judgement.

When it comes to the situation array[head] + array[tail]=x, return two elements in array which add up together is input x.

Which is O(n)

Thus, this algorithm time complexity is O(nlogn)

Persuade:

function find\_two\_elements:

input x, array S

array =merge sort(S)

head = 0

tail= length (S) -1

while head< tail

if array[head] + array[tail]<x:

head +=1

else if array[head] + array[tail]>x:

tail -=1

else: ##array[head] + array[tail]=x

return array[head] , array[tail]

return 0,0

(b)

Apply in hash map

Function find two elements:

Input x, array

For i in length(array):

If array[i] in hashmap:

Return array[i], x-arry[i]

Else:

Add x-array[i] to hashmap

Return -1,-1

this algorithm time complexity is O(n)

2.There are n queries in the question

For each query:

Apply in bisection method to get location about L and R in array;

When it comes to the situation: array[i]<=L<array[i+1]

array[j-1] <R<=array[j], it ends up the binary method.

Return j-i-1 #the number of elements between L and R

Each binary method algorithm is O(logn)

Thus, this question algorithms time complexity is O(nlogn)

3.

In order to get a definitely different choice of team from other people, we can use a question to ask student i whether he choose team i or not:

choice set S

For i in range student number-1:

If student i chose team i, I will not choose team i into my choice set S;

Otherwise I will choose team i, put team number I into S

Return S

this algorithm time complexity is O(n)

4.Apply in bucket sort to think about this question

Consider two special case of this problem.

First one is each point distribute to each bucket,

Every difference maximum is two adjacent bucket size, which is 2\*(1/n). There are n-1 differences.

Thus, the amount of difference is 2(n-1)/n=2-2/n. This value is always below 2.

Second one is one is in the first bucket, and the rest of them are in the second bucket.

In the second bucket, because n-1 points are in this bucket, each one maximum difference is 1/n.

The maximum difference between any point in the other bucket and first bucket’s point is 1.

For this case, the amount of difference is 1+(n-1)/n=2-1/n. This value is always below 2.

Thus, its value is always below 2.

5.The celebrity does not know anyone, but everyone knows him

By using question:

If A knows B:

If answer is yes:

A could not be the candidate, but B may be.

If answer is no:

A may be a possible candidate.

(a)

Step1: First person asks the second one the same question so as to choose possible candidate. Then this possible candidate ask third person same question to choose the person as the new possible candidate. And so on to the end following the same question. To choose a possible candidate. There are n-1 questions.

Step 2: Ask each one among the rest of people, if they know

this possible candidate.

If it occurs any no, there is not a a possible candidate existing.

If it occurs all yes, he may be possible candidate. then go to step3

There are n-1 questions in this case.

Step 3: ask him if he knows each one among the rest of people,

If it occurs any yes, there is not a a possible candidate existing.

If it occurs all no, he must be the candidate.

There are n-1 questions.

Thus, this question need to ask no more than 3n-3 questions.

(b)

Split all the people into a pair of two.

S1: In each pair, the former one asks the latter one.

If former one knows the latter:

If yes, keep the latter one as possible candidate;

If no, keep the former one as possible candidate;

Then adjacent pair’s possible candidate asks the same question in the same way.

In the end choose one possible candidate. There are n-1 questions. There are log2**n** questions involving this candidate.

S2 and S3:

This candidate asks each one among the rest of people,

And each one among the rest of people asks this candidate,

There are 2n-2

But it need to minus log2**n** questions involving this candidate.

Thus, it need to ask no more than 3n - log2**n**-2 questions.

6.

(a) Using the dictionary to record all vote. Set each name on paper as key value and also need to record add 1 into amount.

Record all the paper name.

And then check key values which are names, each amount record just need to decrease 1 in the end, which is this name’s number of vote.

(b) according to question a, whose vote only record 0, which means this kind of person is the students who did not

receive any votes.

Use that name to check all the paper.to find other name on the paper which occurs that name. And we can get the people who these student vote for.

(c) there are n voters to vote. We can know that each one can vote once. There are n vote papers. And every student received at least one vote.

Thus, in the end, there is no any rest vote to distribute others, the

maximum possible number of votes received by any student is 1.

(d)

Time complexity: O(n)

we can use no direction graph, queue, set, BFS method

first, we need to build graph

step1:

build graph:

set every vote as the edge, such as

edge(name 1,name 2),

every name is a vertex.

Put vertexes of names into the set

When each name occurs once, this vertex’s degree +1

Calculate all the vertexes which form by names and connect all the names which appear on the same paper as edges

Step2:

Using BFS traverse all vertexes and enqueuer the vertexes whose degree are 1 into queue.

Then dequeuer the element in queue:

Every time dequeue one element A and remove element A from set and get the edge(A,B) of the graph and then print this edge also and then decrease one degree of the element B which is connect to A

and then check if B’s degree is equal to 1:

if yes, add B into queue;

if no, dequeue the next element in the queue until queue is empty.

Step 3:

To check the set is empty or not; if not print in the while loop;

If set:

If yes, return to end

If no, goes to while loop

While next vertex is not stop # set set[0] as one starting vertex C of edge(C,D),find D as the next vertex and other edge is (C,F)

When next becomes to F, it goes to end. Special case is there are (C,D) and (D,C). #in order to get the loop direction of vote.

C D

F E

While next is not equal to stop:

print edge (vertex, next)

vertex= next

next vertex change to vertex (next , next\_next)