

No Books or Photocopies.

Weightage: 10%, Max Marks - 40

1. Recall the quick-find algorithm discussed in class (also given below, for your reference). An input sequence 0-2, 1-4, 2-5, 3-6, 0-4, 6-0 and 1-3 is given.
 - (a) Show the contents of the **id** array after each *union* operation. (2 Marks)
 - (b) Give the number of times the program accesses the array for each input pair. (2 Marks)
 - (c) Explain the computational complexity of the quick-find algorithm. Think about how the algorithm would work for arbitrary input pair sequences. (2 Marks)
 - (d) Give an example of a best-case input sequence and an example of a worst-case input sequence, for this algorithm. (4 Marks)

```
#include<iostream.h>
static const int N=10000;
int main()
{
    int i, p, q, id[N];
    for(i=0; i<N; i++) id[i] = i;
    while(cin >> p >> q)
    {
        int t = id[p];
        if (t == id[q]) continue;
        for(i=0; i<n; i++)
            if(id[i] == t) id[i] = id[q];
        cout << " " << p << " " << q << endl;
    }
}
```

(The purpose of this question is to help you test your own understanding of (i) hand-execution process and (ii) computational complexity calculations.

You will be evaluated on: (i) The correctness of your answers, and (ii) Neatness and readability of your work.)

2. Recall the leader election algorithm (Josephus problem) that was enacted in class. The problem is repeated here (for those do not recall it): Imagine that N people have decided to elect a leader by arranging themselves in a circle and eliminating every Mth person around the circle, till only one person remains, who becomes the leader.

Now suppose you want the person at the Kth position to become the leader.

- (a) Design an algorithm that will output M, given N and K. Clearly state any assumptions that you make and justify them. (3 Marks)
- (b) Show the pseudo-code to implement the above algorithm. (5 Marks)
- (c) Show the working of your algorithm when N=7 and K=3. (2 Marks)

(The purpose of this question is to help you test your own understanding of how well you are able to modify something that you have already learnt.

You will be evaluated on: (i) The correctness of your algorithm and pseudo-code, (ii) The justification for the assumption and any other design choices that you may make, (iii) Neatness and readability of your work.)

3. Suppose that a small company has given your group the contract of building a request-response application server. This means that the server should take an arriving request as input, process the request, and return the response as output. Some typical examples of such applications are searching on Google, accessing any database and railway booking counter.

Assume that your team has 3 people and the work has been divided as follows: You are supposed to design the data structure, its interfaces and implementation, for handling the incoming requests. Another person will take the request(s) from your data structure, do the necessary processing and create the output. The third person will return the output as being the response corresponding the given request.

You only know about arrays and linked-lists (Recall the insertion sort algorithm that was enacted in class). You have to construct your new data structure based on your knowledge of arrays and/or linked-lists.

- (a) Draw a picture of your data structure, and illustrate its use with an example. This will help you to figure out its interface and implementation. **(2 Marks)**
- (b) Write the code for its interface. In other words, show the struct and function definitions. **(2 Marks)**
- (c) Write the code for its implementation. In other words, show how the functions defined above will work. **(2 Marks)**
- (d) Write a small client program that you can use to test whether your implementation works correctly or not. **(2 Marks)**
- (e) Determine the computational complexity of your client program. **(2 Marks)**

(The purpose of this question is to help you to test your own understanding of (i) the software engineering process discussed in class, and (ii) whether you can extend your knowledge and build new data structures from familiar ones.

You will be evaluated on: (i) Have you listed all the necessary assumptions regarding the requirements, along with their justification, (ii) How do you evaluate whether to choose an array or a linked-list as the basis for your data structure, and justification of your choice, (iii) Does your pseudo-code specify all the necessary informaton required for writing the implementation code of your data structure, (iv) What client program and inputs that you use to test the correctness of your code, (v) What reasoning do you use to determine the computational complexity, (vi) Neatness and readability of your work.)

4. Attach your 2 A4 handwritten sheets to this answer paper. **(1 Mark)**

5. Take Home (Individual assignment): Find a computer to implement and test your solutions to Questions 2 and 3 above. Submit your code and test cases used, on Moodle, by 9 pm on Tuesday (9th Feb). Check the Moodle page for instructions from the TAs on how you should name your files etc. **(10 Marks)**

(The purpose of this question is for you to see for yourself whether your pseudo-code is correct or not, by actual implementation.

NOTE:

- (a) You should not change the data structure that you used for the pseudo-code during the quiz. For example, if your pseudo-code submitted was using arrays, your implementation should also be using arrays. You may fix bugs in your psuedo-code logic to ensure that the implementation works correctly.
- (b) Sophisticated copy-detection software is being used. If A has copied from B, with or without B's knowledge, both A and B will get Zero for the entire Quiz. Absolutely no cribs or explanations will be entertained.

)
