Quiz Instructions

Please answer all questions. Please answer in the format requested in the question. For example, if the question says "list the node number separated by spaces", then please don't answer "Nodes 1, 2, 3, 4". Instead, answer "1 2 3 4".

In the programming questions, you do not need to put "using namespace std" or any "#include" statements.

The point values are guidelines for the number of minutes that it should take you to do each question.

I will not be grading this on Canvas -- Instead, you will receive your grade via your "Grappling Hook" assignment.

Question 1 15 pts

Part A: You force each of your employees to choose a password composed of exactly 5 lower-case letters followed by 4 single numeric digits. How many potential passwords are there?

Part B: You are running a lottery. You have a machine with 17 ping-pong balls, numbered 1 to 17. It has been programed to choose one of these balls at random every second. Each time a ball is chosen, it is put back onto the machine, so your machine always has the same 17 balls. Your lottery number is composed of the ping-pong ball numbers chosen in the last 6 seconds, ordered from most recently-chosen to least recently-chosen. How many different lottery numbers are there?

Part C: Yo Your new I How many	home	e or	ly ha	s ro	om	fo	r 12	2 TV	s, s	y yc	ou h	av	e to	do	ona	te \$	5 of	f th	em	to (nrity
Part D: The vertically. 11 different sure that the nights car.	You nt-co the g	car lore els	put a ed gel form	a ge s. E a d	l o ac iffe	ver h n ren	a I igh it a	ight, t, yo rrang	and u'll u gem	tha use ent	at n all of	nak of t	the	it a ge ev	asp Is,	eci but	ific yo	col u w	or. van	You t to	ı h ma	ave
Part E: I'm of juice I whad. She should be combine a altogether wrong. How is):	vant. said, any o ! See	Ins "I h f the e, th	tead of ave 1 em the ere a	of ju 2 ki at y re ir	st s nd: ou nfin	say s of wa iite	ing f jui nt i cor	"Ora ce. ` n eq mbin	ange You o ual o atior	", I can juar is!!'	wa ang ntiti ' Yo	s do y of es. ou're	uml f the Or e to	b e em yo o p	nou on u ca oolit	igh the an s	to eir d skip o te	ask own o ju ell h	k wl n, on ice er t	nat ryo that	she u c	e an e's
Questic			ing is	an	ins	tan	ce	of di	sjoin	t se	ets:									1	5 p	ts
Node:	0 1	2						9 1				14	15	16	17	18	19	20	21	22	23	2
4 Links: 1	1 -1	-1	10 21	-1	5	-1	18	24 2	4 7	5	7	1	1	24	0	5	6	24	7	0	1	-
Please en							ode	es in	the	san	ne s	set	as I	noc	de (5. F	Ple	ase	ju:	st e	nte	r

Part B: The following is an instance of disjoint sets:

Node: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 2 4 Links: -1 20 20 6 21 6 -1 15 -1 23 0 6 15 5 1 21 6 5 0 16 -1 6 6 -1 1

- Please enter the return value of Find(7):
- Please enter the return value of Find(18):
- Please enter the return value of Find(28):
- Please enter the return value of Find(14):

Part C: The following is an instance of disjoint sets:

Node: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 2 4 Links: 17 5 8 22 19 17 16 18 -1 19 8 13 17 -1 17 8 8 8 8 -1 8 19 18 -1 8 -1 1

Suppose we are using Union-By-Rank-With-Path-Compression. When we call Find(1), which node(s) change their Links field (just put node numbers separated by spaces)?

For any of the link fields in the previous question, what is the new value (just answer one)?

Question 3 1.5 pts

In the following procedure, suppose that the map's size is *n* and that the vector's size is *m*. What is the running time of the following procedure:

```
void p_b8ca2(const map <int, int> &x, vector <int> &y)
{
  map <int, int>::iterator xit;
  for (xit = x.begin(); xit != x.end(); xit++) {
    y.push_back(xit->first);
    y.push_back(xit->second);
}
```

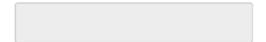
Your answer should be a big-O function of *n* and/or *m*.

Question 4 1.5 pts

In the following procedure, suppose that the map's size is n and the vector's size is m. What is the running time of the following procedure:

```
int p_07659(map <int, int> &x, vector <int> &v)
{
   int nfound, index;

   index = v.size()-1;
   nfound = 0;
   do = {
      if (x.find(v[index]) != x.end()) nfound++;
      index /= 2;
   } while (index != 0);
   return nfound;
}
```



Question 5 1.5 pts

In the following procedure, suppose that the set's size is *n* and that its largest element is *m*. What is the running time of the following procedure:

```
unsigned long long p_77587(const set <int> &s1)
{
    set <int>::iterator sit;

    unsigned long long s2;

    s2 = 0;
    for (sit = s1.begin(); sit != s1.end(); sit++) {
        s2 l= (1 << (*sit));
    }
    return s2;
}</pre>
```

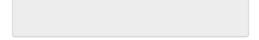
Your answer should be a big-O function of *n* and/or *m*.

Question 6 1.5 pts

In the following procedure, suppose that the set's size is n and that the vector's size is m. What is the running time of the following procedure:

```
int p_d9450(const set <int> &x, const vector <int> &y)
{
   int n;
   size_t i;
   set <int>::iterator xit;

n = 0;
   for (i = 0; i < y.size(); i++) {
      xit = x.lower_bound(y[i]);
      if (xit != x.end()) n += (*xit - y[i]);
   }
   return n;
}</pre>
```



Question 7 1.5 pts

In the following procedure, suppose that the disjoint set has *n* elements and the vector has *m* elements. The implementation of disjoint sets is Union-By-Rank-With-Path-Compression. What is the running time of the following procedure:

```
void p_1ea1f(Disjoint_Set &x, vector <int> &set_ids)
{
    size_t i;

    for (i = 0; i < set_ids.size(); i++) {
        set_ids[i] = x.Find(set_ids[i]);
    }
}</pre>
```

Your answer should be a big-O function of *n* and/or *m*.

Question 8 1.5 pts

In the following procedure, suppose that the map's size is n and that its largest element is m. What is the running time of the following procedure:

```
void p_c3b36(map <int, int> &x)
{
  int n;

  n = x.size();
  while (x.size() > 3*n/4) x.erase(x.begin());
}
```

Question 9 1.5 pts

In the following procedure, suppose that the disjoint set has *n* elements and the vector has *m* elements. The implementation of disjoint sets is Union-By-Rank-With-Path-Compression. What is the running time of the following procedure:

```
void p_014aa(Disjoint_Set &x, const vector <int> &set_ids)
{
    size_t i;

    for (i = 0; i < set_ids.size(); i += 2) {
        x.Union(set_ids[i], set_ids[i+1]);
    }
}</pre>
```

Your answer should be a big-O function of n and/or m.

Question 10 1.5 pts

In the following procedure, suppose that s1's size is *n* and that s2's size is *m*. What is the running time of the following procedure:

```
size_t p_c3b36(const string &s1, const string &s2)
{
    size_t i, found;

    found = 0;
    for (i = 0; i < s1.size(); i++) {
        if (s2.find(s1[i]) != string::npos) found++;
    }
}</pre>
```

Question 11 1.5 pts

In the following procedure, suppose that the vector's size is *n* and that the multiset starts empty. What is the running time of the following procedure:

```
void p_5a7e8(const vector <int> &v, multiset <int> &s)
{
    size_t i;
    for (i = 0; i < v.size(); i++) s.insert(v[i]);
}</pre>
```

Your answer should be a big-O function of n.

Question 12 1.5 pts

In the following procedure, suppose that v's size is n and that rv's size is m. What is the running time of the following procedure:

```
void p_37089(const vector <int> &v, vector <int> &rv)
{
  int i, j;

  for (i = 0; i < v.size(); i++) {
    for (j = i+1; j < v.size(); j++) {
      rv.push_back(v[i] * v[j]);
    }
}</pre>
```

Question 13 8 pts

Your goal here is to write the recursive procedure binstrings(), which has the following prototype:

```
void binstrings(int n, int index, string &sofar);
```

It will be called in the main() below. It should print all *n*-letter strings composed of the characters 'A' and 'B'. You must use recursion to perform this task. You are not allowed to have a for, do or while loop in this program.

```
int main()
{
  int n;
  string s;

cin >> n;
  s.resize('-', n);
  binstrings(n, 0, s);
  return 0;
}
```



Question 14 8 pts

Your goal here is to write the procedure enum_cd(), which has the following prototype:

```
void enum_cd(int n);
```

It will be called in the main() below. It should print all *n*-letter strings composed of the characters 'C' and 'D'. You should use one of the enumerations that you learned in this class to perform this task.

```
int main()
{
  int n;
  cin >> n;
  enum_cd(n);
  return 0;
}
```



Question 15 14 pts

Behold the header file **collection.hpp**:

```
#pragma once
#include <string>
using namespace std;

class Collection {
  public:
    virtual ~Collection() {};
    virtual void Add_Item(int id, const string &thing) = 0;
    virtual void Print() = 0;
};
```

A collection stores items. Each item has an integer id, and then some collection of "things". Each thing is defined by a string, and you can have multiple things with the same string. The methods are defined as follows:

- Add_Item() adds the item with the given id to the collection, if it's not there
 already. It then adds the string "thing" to the item's collection of strings.
- Print() prints out the items, one per line. To print an item, it should print the item's
 id followed by a colon (no space between the id and the colon). It should then
 print the "things" that belong to the item, each separated by a space. It doesn't
 matter what order you print out the items or the things.

You are going to implement the class **MyImp**, which will implement the **Collection** interface. Do so below, first by specifying the **MyImp** class as you would in a header file (**myimp.hpp**), and then by implementing the methods as you would in an implementation file (**myimp.cpp**).

Here's an example main();

```
#include <iostream>
#include "collection.hpp"
#include "myimp.hpp"
using namespace std;

int main()
{
   int id;
   string item;
   Collection *c;

   c = new MyImp;

   while (cin >> id >> item) c->Add_Item(id, item);
   c->Print();
   return 0;
}
```

And here's an example of it compiled to a.out and running:

```
UNIX> cat items.txt
23894 dog
23894 cat
854 cat
23894 cat
7 fred
7 binky
UNIX> a.out < items.txt
7: fred binky
854: cat
23894: dog cat cat
UNIX>
```

I'm giving you flexibility in how you implement this, so although that's a legal output above, there are many other legal outputs. For example the following outputs are also legal:

```
UNIX> a.out < items.txt
23894: dog cat cat
7: fred binky
854: cat
UNIX>
```

or

854: cat	at dog cat	txt			
	to implemen		-	on and is effic	cient.