COSC202 Midterm Exam - 11/6/2025 - James S. Plank

Please answer all questions -- put your answers here on the exam.

Question 1 - Straightforward Big-O. ---------- Name/Username: _____

For each procedure, please put the best big-O running time of the procedure. Just write it in an empty space in the box.

Part A:

```
void pa91cd(int n)
{
  int i;
  deque <int> v;

  for (i = 0; i < n; i++) v.push_back(i);
}
```

Part B:

```
void pc66fb(int n)
{
  int i;
  list <int> v;

  for (i = 0; i < n; i++) v.insert(v.begin(), i);
}
```

Part C:

```
void pcd698(int n)
{
  int i;
  set <int> v;

  for (i = 0; i < n; i++) v.insert(i);
}
```

Part D:

```
void p5f687(int n)
{
  int i;
  map <int, int> v;

  for (i = 0; i < n; i++) v.insert(make_pair(i,i));
}
```

Part E:

```
void p100e0(int n)
{
  int i;
  deque <int> v;

  for (i = 0; i < n; i++) v.push_front(i);
}
```

Part F:

```
void p9edbf(int n)
{
  int i;
  unordered_map <int, int> v;

  for (i = 0; i < n; i++) v.insert(make_pair(i,i));
}
```

Part G:

```
int p8326b(int n)
{
  int i, s;

  s = 0;
  for (i = 0; i < (1 << n); i++) s++;
  return s;
}
```

Part H:

```
void p0669d(int n)
{
  int i;
  vector <int> v;

  for (i = 0; i < n; i++) v.push_back(i);
}
```

Part I:

```
int p6f3cf(int n)
{
  int i, j, s;

  s = 0;
  for (i = 0; i < n; i++) {
    for (j = 0; j < i; j++) s++;
  }
  return s;
}
```

Part J:

```
void pab7da(int n)
{
  vector <int> v;
  int i;

  for (i = 0; i < n; i++) v.insert(v.begin(), i);
}
```

Question 2 - More challenging big-O. ---------- Name/Username: _____

For each procedure, assume that **v1** has *n* elements, and if it is a vector of vectors, that it is a vector of *n* vectors of *n* integers. Please answer the big-O running time of the procedure. Just write the answer in an empty space in the box. The answer will be a function of *n*, a function of *m* or a function of both.

Part A:

```
int p52723(map <int, int> &v1, int m)
{
  int i, total;
  map <int, int>::iterator mit;

  total = 0;
  for (i = 0; i < m; i++) {
    for (mit = v1.begin(); mit != v1.end(); mit++) {
      total += (i + mit->first - mit->second);
    }
  }
  return total;
}
```

Part B:

```
int p5cb3c(set <int> &v1, int m)
{
  int i;
  int j;

  j = 0;
  for (i = 0; i < m; i++) {
    if (v1.find(i) != v1.end()) j++;
  }
  return j;
}
```

Part C:

```
map <int, int> pf1032(int n)
{
  int i;
  map <int, int> rv;

  for (i = 0; i < n; i++) {
    rv.insert(make_pair(1, i));
  }
  return rv;
}
```

Part D:

```
int pfb194(int n, int m)
{
  int i, j, s;

  s = 0;
  for (i = 0; i < n; i++) {
    for (j = 1; j < m; j = 2 * j) s++;
  }
  return s;
}
```

Part E:

```
int p2c465(vector < vector < int> > v1)
{
  int i, j, t, n;

  n = 0;
  t = 0;
  for (i = 0; i < v1.size(); i += 10) {
    for (j = 0; j < v1[i].size(); j += 5) {
      t += v1[i][j];
      n++;
    }
  }
  return t/n;
}
```

Part F:

```
vector <int> p7ad0a(vector <int> &v1)
{
  vector <int> rv;
  int i, k, t, n;

  for (i = 1; i < v1.size()-1; i++) {
    t = 0;
    n = 0;
    for (k = -1; k <= 1; k++) {
      n++;
      t += v1[i+k];
    }
    rv.push_back(t/n);
  }
  return rv;
}
```

Part G:

```
int pabfda(vector <int> &v1, int m)
{
  int i, j, t;

  j = 1;
  t = 0;
  for (i = 0; i < v1.size(); i++) j *= 2;
  for (i = 0; i < j; i++) {
    t += (v1[j] % v1.size());
  }
  return t;
}
```

Part H:

```
void p8099b(vector <int> &v1, int m)
{
  int i;

  for (i = 0; i < m; i++) v1.push_back(i);
}
```

Question 3 ---------- Name/Username: _____

Below are class definitions for **Stack**, **Queue** and **Dlist**, from the lecture notes. I have modified the definitions so that everything is **public**, and I have omited the constructors, destructors, assignment overloads, and other methods that are irrelevant to this problem.

```
class Stacknode {
  public:
    std::string s;
    Stacknode *next;
};

class Stack {
  public:
    bool Empty() const;
    size_t Size() const;

    void Push(const std::string &s);
    std::string Pop();

    Stacknode *top;
    size_t size;
};
```

```
class Qnode {
  public:
    std::string s;
    Qnode *ptr;
};

class Queue {
  public:
    void Clear();
    bool Empty() const;
    size_t Size() const;

    void Push(const std::string &s);
    std::string Pop();

    Qnode *first;
    Qnode *last;
    int size;
};
```

```
class Dnode {
  public:
    std::string s;
    Dnode *Next();
    Dnode *Prev();
    Dnode *flink;
    Dnode *blink;
};

class Dlist {
  public:
    void Clear();
    bool Empty() const;
    size_t Size() const;

    void Push_Front(const std::string &s);
    void Push_Back(const std::string &s);

    std::string Pop_Front();
    std::string Pop_Back();

    Dnode *sentinel;
    size_t size;
};
```

For each program below, please tell me the output. Note, I have labeled the **cout** statements with numbers -- put the output of that **cout** statement on the line with the proper number. If you don't know the output, then put "unknown." If there is a segmentation violation, the put "segfault", but assume that the program continues to run as if the segmentation violation never occurred.

```
int main()
{
  Stack s;
  Stacknode *sn;
  int t;
  string x;

  s.Push("Aidan");
  s.Push("Blake");
  s.Push("Claire");
  s.Push("David");
  x = s.Pop();
  cout << x << endl;                 // 1
  cout << s.top->s << endl;          // 2
  cout << s.top->next->s << endl; // 3

  t = 0;
  for (sn = s.top;
       sn != NULL;
       sn = sn->next) t++;

  cout << t << endl;                 // 4

  return 0;
}









// 1. _____________________________
//
// 2. _____________________________
//
// 3. _____________________________
//
// 4. _____________________________
```

```
int main()
{
  string s;
  Queue q;
  Qnode *qn;
  int t1, t2;

  q.Push("Elijah");
  q.Push("Feliciano");
  q.Push("Gregory");
  q.Push("Hannah");
  s = q.Pop();

  cout << s << endl;              // 5
  cout << q.first->s << endl;     // 6
  cout << q.last->s << endl;      // 7

  t1 = 0;
  for (qn = q.first;
       qn != NULL;
       qn = qn->ptr) t1++;
  cout << t1 << endl;             // 8

  t2 = 0;
  for (qn = q.last;
       qn != NULL;
       qn = qn->ptr) t2++;
  cout << t2 << endl;             // 9

  return 0;
}

// 5. _____________________________
//
// 6. _____________________________
//
// 7. _____________________________
//
// 8. _____________________________
//
// 9. _____________________________
```

```
int main()
{
  Dlist d;
  Dnode *dn;
  int t;
  string s;

  d.Push_Back("Ingrid");
  d.Push_Back("James");
  d.Push_Front("Kathleen");
  d.Push_Front("Logan");
  d.Push_Front("Madison");
  s = d.Pop_Back();

  cout << s << endl;                                  // 10
  cout << d.sentinel->flink->s << endl;               // 11
  cout << d.sentinel->flink->flink->s << endl;        // 12
  cout << d.sentinel->blink->s << endl;               // 13
  cout << d.sentinel->blink->blink->s << endl;        // 14

  t = 0;
  for (dn = d.sentinel;
       t == 0 || (t < 100 && dn != d.sentinel);
       dn = dn->flink) t++;
  cout << t << endl;                                  // 15

  return 0;
}


// 10. _____________________________
//
// 11. _____________________________
//
// 12. _____________________________
//
// 13. _____________________________
//
// 14. _____________________________
//
// 15. _____________________________
```

Question 4 ---------- Name/Username: _____

Below is a header file for a data structure holding a collection of people:

```
#include <iostream>
#include <unordered_map>
#include <vector>
#include <map>
using namespace std;

class Person {
  public:
    string name;
    int age;
};

class People {
  public:
    People();
    ~People();
    People(const People &p);
    People& operator= (const People &p);

    bool add_person(const string &name, int age);  // Add the given person to the collection.  If the person's
                                                   // name is already there, return false and do not update
                                                   // the age.  Otherwise return true.

    void print_by_age();                           // Print all people ordered by age.  For each person,
                                                   // print a line with the age, a space, and then the person's name.
                                                   // Don't worry about order if two people have the same age.
  protected:
    vector <Person *> pvec;
    unordered_map <string, Person *> pmap;
};
```

Question 3A: Which of the methods above should be labeled with **const**? _____

Question 3B: Which of the six implementations of the destructor below is the best? Just circle it or put a big check mark in the box with the best implementation.

```
People::~People()
{
  delete pvec;
  delete pmap;
}
```

```
People::~People()
{
  size_t i;
  unordered_map <string, Person *>::iterator mit;

  for (i = 0; i < pvec.size(); i++) delete(pvec[i]);
  for (mit = pmap.begin(); mit != pmap.end(); mit++) {
    delete mit->second;
  }
}
```

```
People::~People()
{
  size_t i;
  Person *p;

  for (i = 0; i < pvec.size(); i++) {
    p = pvec[i];
    delete(pmap[p->name]));
  }
}
```

```
People::~People()
{
  pvec.clear();
  pmap.clear();
}
```

```
People::~People()
{
  return;    // The default constructor works here.
}
```

```
People::~People()
{
  size_t i;

  for (i = 0; i < pvec.size(); i++) delete(pvec[i]);
}
```

Questions 5 and 6 ---------- Name/Username: _____

Question 5: Implement the **add_person()** method from the header file on the previous page. Please assume you have included the header file.

```
bool People::add_person(const string &name, int age)
{

}
```

Question 6: Implement the **print_by_age()** method from the header file on the previous page.

```
void People::print_by_age()
{

}
```