

# CS302 Midterm Exam

James S. Plank - September 30, 2010

Answer all questions.

Please put all of your answers on one or more separate sheets of paper and hand those in.  
Put your name and email address on all sheets.

Do not staple them together, please.

Please do not, and I mean do not write answers on this exam and hand that in! Thanks.

## Question 1

**Part 1:** For each of the three programs below, explain *precisely* what they do. Don't tell me that program A pushes values onto a vector. Instead, tell me what its output is in terms of its input.

**Part 2:** For each of the three programs below, state its big-O running time in terms of its input. Again, be as *precise* as possible. And let me reiterate. Be as *precise* as possible.

A:

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

main()
{
    vector <int> iv;
    int i;

    while (cin >> i) iv.push_back(i);
    for (i = 0; i < iv.size(); i++) {
        cout << iv[i] << endl;
    }
}
```

B:

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

main()
{
    vector <int> iv;
    int i;

    while (cin >> i) iv.insert(iv.begin(), i);
    for (i = 0; i < iv.size(); i++) {
        cout << iv[i] << endl;
    }
}
```

C:

```
#include <iostream>
#include <set>
using namespace std;

main()
{
    set <int> iv;
    set <int>::iterator vit;
    int i;

    while (cin >> i) iv.insert(i);
    for (vit = iv.begin(); vit != iv.end(); vit++) cout << *vit << endl;
}
```

## Question 2

Give me an example of operator overloading in C++.

## Question 3

The procedure to the right calculates a weighted average of a vector of doubles. Explain why it would be better to rewrite this procedure using a reference parameter, and then rewrite it so that it uses one.

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

double weighted_average(vector <double> v)
{
    int i;
    double total;

    for (i = 0; i < v.size(); i++) v[i] *= ((double) (i+1));
    total = 0;
    for (i = 0; i < v.size(); i++) total += v[i];
    if (v.size() == 0) return 0;
    return total / (double) v.size();
}
```

### Question 4

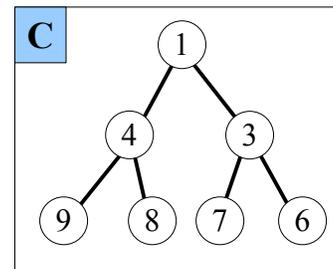
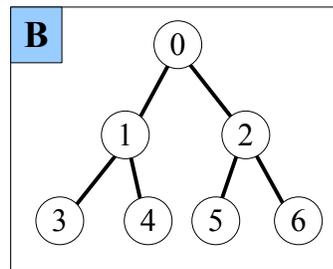
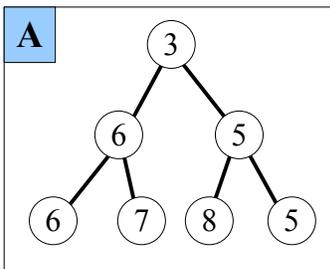
Write a procedure `is_heap()` with the following prototype:

```
int is_heap(vector <double> &h);
```

The procedure should return 1 if `h` is the vector representation of a valid heap. It should return 0 otherwise.

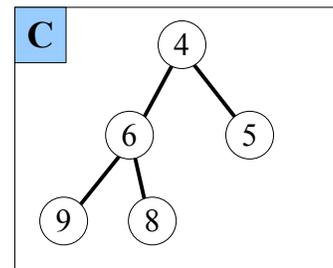
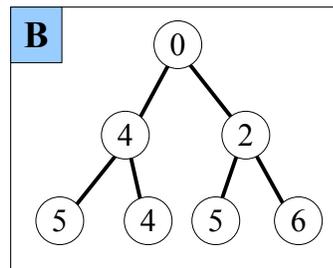
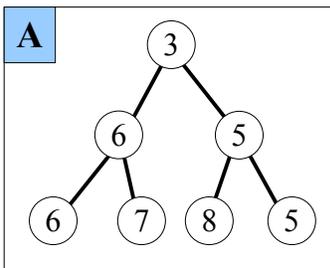
### Question 5

For each heap below, draw the result of calling `Pop()` on the heap. In each case, what is the return value of `Pop()`:



### Question 6

For each heap below, draw the result of calling `Push(3)` on the heap.



### Question 7

We can implement the priority queue data structure with a C++ multiset or with a heap. Give me two reasons why using a heap is better than using a multiset.

### Question 8

What desirable property does insertion sort have that bubble sort and selection sort do not have?

## **Question 9**

Write insertion sort with the following prototype. You get extra points if you use a sentinel so that it is faster than an implementation that doesn't use a sentinel.

```
void insertion_sort(vector <double> &v);
```

## **Question A**

Suppose you are sorting the vector of strings to the right. Show me what the vector will look like after every pass during:

- The first two passes of bubble sort
- The first three passes of selection sort
- All passes of insertion sort.

```
E H B F A U G M S P
```

You may assume that we are not using the sentinel with insertion sort.

## **Question B**

Write a program that prints out all 12-letter strings composed of the letters A and B. Hint: Use bit arithmetic.

## **Prototypes from the Standard Template Library**

```
Vectors:  
  
void push_back( const TYPE& val );  
iterator begin();  
size_type size() const;  
iterator insert( iterator loc, const TYPE& val );  
  
Sets:  
  
iterator begin();  
iterator end();  
pair<iterator, bool> insert(const TYPE& val );
```