CS 7B - Spring 2018 - Coding an Algorithm Quiz Solutions

Consider the following precondition and postcondition specifications:
Precondition A list of n different integers: $\{x_1, x_2, \ldots, x_n\}$ so that $x_i \neq x_j$ if $i \neq j$
Postcondition The same list of numbers sorted in increasing order.

For example, the list $\{2, 1, 5, 3\}$ could be an input (precondition) and then the postcondition would be $\{1, 2, 3, 5\}$.

1. Write pseudocode for an algorithm that will accomplish this.
   Did I say "sort"? Oh dear...that's potentially complicated! I had a simpler notion in mind, but the sort thing just kinda popped out! Oh, hell...so here it goes:
   There are many, many sort routines. Most computer time is spent searching and sorting; sorting makes it easier to search...

   Here's a link to a fairly comprehensive collection of sorting algorithms: https://www.toptal.com/developers/sorting-algorithms. Let's pick selection sort, whose algorithm in pseudo code might look like this:

```
for i = 2:n,
    for (k = i; k > 1 and a[k] < a[k-1]; k--)
        swap a[k,k-1]
     invariant: a[1..i] is sorted
end
```

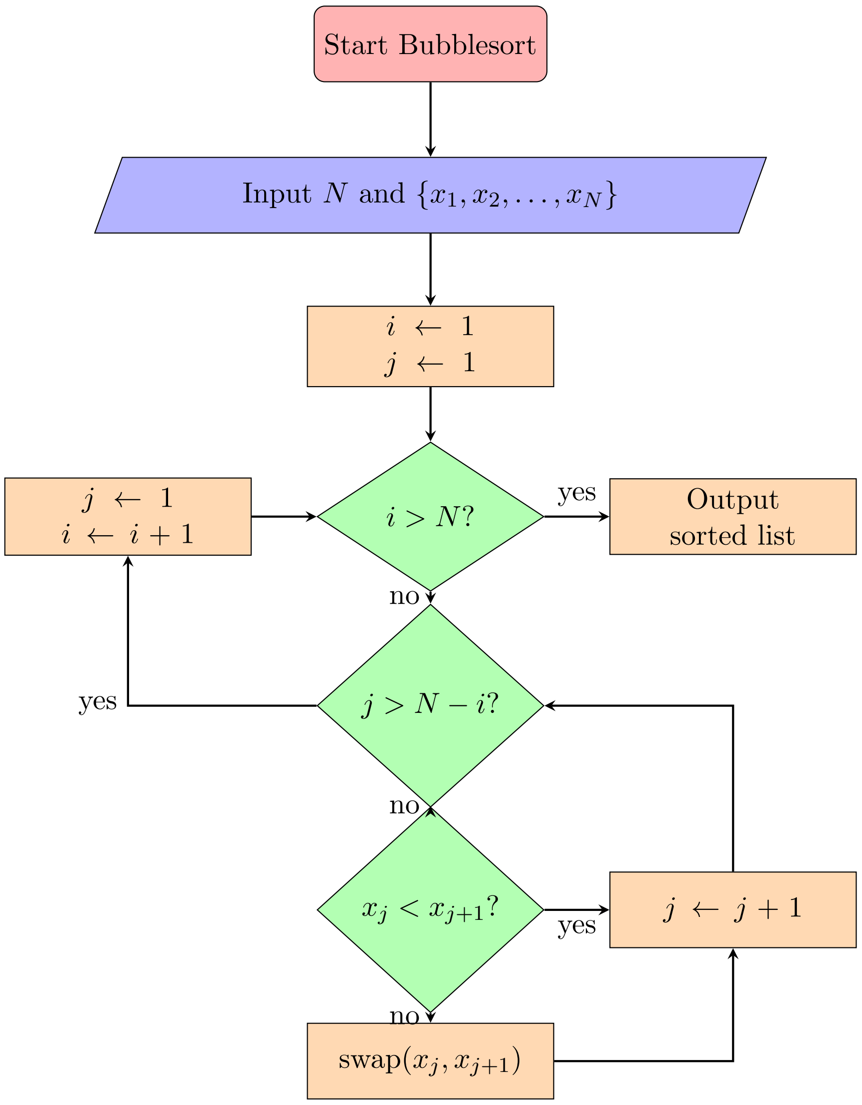
   Ok, I take it back. Let's do bubblesort. The aforementioned sorting web site has this algo:

```
for i = 1:n,
    swapped = false
    for j = n:i+1,
        if a[j] < a[j-1],
            swap a[j,j-1]
            swapped = true
     invariant: a[1..i] in final position
    break if not swapped
end
```

   That's no doubt improved over mine:

```
for i = 1 to N,
    for j = 1 to N-i
        if x[j]>x[j+1]
            swap(x[j],x[j+1])
```

2. Sketch a flow-chart for your algorithm.
   Here you go:



3. Write C++ code that will implement your algorithm.

```
#include <iostream>
#include <vector>
#include <cstdlib>
#include <ctime>

using namespace std;

/**
for i = 1 to N,
    for j = 1 to N-i
        if x[j]>x[j+1]
            swap(x[j],x[j+1])
*/

int bubbleSort(vector<int>& v) {
    int swaps{0};
    for(int i = 0; i < v.size(); ++i)
        for(int j = 0; j < v.size()-i-1; ++j)
            if(v[j]>v[j+1]) {
                swap(v[j],v[j+1]);
                ++swaps;
            }
    return swaps;
}

void show(vector<int>& v) {
    for(int i:v) cout << i << " ";
}
```

```
void initializeV(int N, vector<int>& v, bool randm = 1) {
    if(randm) {
        for(int i = 0; i < v.size(); ++i) v[i]=(rand()%10000);
    }
    else for(int i = 0; i < N; ++i) {
        cout << "\nEnter c[" << i << "] = ";
        cin >> v[i];
    }
}

int main() {
    int N{100};
    vector<int> v(N);
    int sumSwaps{0};
    for(int n = 0; n < 1000; ++n) {
        initializeV(N,v);
        sumSwaps += bubbleSort(v);
    }
    cout << "\nThe percentage of swaps = " << double(sumSwaps)/(1000*N);
    return 0;
}
```

4. What is an invariant in your code?
   After iteration `i` the values $\{x_{N-i}, x_{N-i+1}, \ldots, xN\}$ will contain the `i` largest values of the list in ascending order.

5. What is the number of comparisons your code requires for a list of n different integers?
   The `i`th pass through the outer loop requires $N-i$ comparisons, so there are a total of $N+(N-1)+(N-2)+\cdots+1 = \frac{N(N-1)}{2}$ comparisons.

6. What is the average number of times your algorithm will swap two numbers for a list of three integers?
   The code above gives a general impirical result of about 24%.