**EE2310 C++程式設計HW 6 (Chapter 9) due: 5/28/2020**

**Part 1, Choice(s) 選擇題 (30%)**

1) The advantage of a linear search is that \_\_\_\_\_\_\_\_.

A) it is simple

B) it is efficient

C) it is fast

D) it can be used on unordered data

E) both A and D

2) A(n) \_\_\_\_\_\_\_\_ search is more efficient than a(n) \_\_\_\_\_\_\_\_ search.

A) string, double

B) integer, double

C) binary, linear

D) linear, binary

E) None of the above; all searches are equally efficient.

3) The linear search is adequate for searching through \_\_\_\_\_\_\_\_ arrays, but not through \_\_\_\_\_\_\_\_ ones.

A) int, double

B) char, string

C) ascending, descending

D) small, large

E) any regular, vector

4) A binary search begins by examining the \_\_\_\_\_\_\_\_ element of an array.

A) first

B) last

C) largest

D) middle

E) smallest

5) If the item being searched for is not in the array, binary search stops looking for it and reports that it is not there when \_\_\_\_\_\_\_\_.

A) array index first> array index last

B) Boolean variable found= false

C) Boolean variable found= true

D) it finds a value larger than the search key

E) it has examined all the elements in the array

6) A search can be performed on an array of \_\_\_\_\_\_\_\_ .

A) integers

B) strings

C) objects

D) All of the above

E) A and B, but not C

7) A sorting algorithm can be used to arrange a set of \_\_\_\_\_\_\_\_ in \_\_\_\_\_\_\_\_ order.

A) numeric values, ascending

B) numeric values, descending

C) strings, ascending

D) strings, descending

E) All of the above

8) The \_\_\_\_\_\_\_\_ sort usually performs fewer exchanges than the \_\_\_\_\_\_\_\_ sort.

A) bubble, selection

B) selection, bubble

C) binary, linear

D) linear, binary

E) linear, bubble

9) To find a value that is in an unordered array of 100 items, linear search must examine an **average** of \_\_\_\_\_\_\_\_ values.

A) 7

B) 10

C) 50

D) 100

E) 101

10) To determine that an item is not in an unordered array of 100 items, linear search must examine an **average** of \_\_\_\_\_\_\_\_ values.

A) 7

B) 10

C) 50

D) 100

E) 101

11) To locate a value in an ordered array of 100 items, binary search must examine **at most** \_\_\_\_\_\_\_\_ values.

A) 7

B) 10

C) 50

D) 100

E) 101

12) A bubble sort is being used to arrange the following set of numbers in ascending order:

7 5 3 9 2 6

After the first pass of the sort is completed, what order will the data be in?

A) 2 5 3 9 7 6

B) 5 7 3 9 2 6

C) 5 3 7 2 6 9

D) 2 3 5 6 7 9

E) None of the above

13) A selection sort is being used to arrange the following set of numbers in ascending order:

7 5 3 9 2 6

After the first pass of the sort is completed, what order will the data be in?

A) 2 5 3 9 7 6

B) 5 7 3 9 2 6

C) 5 3 7 2 6 9

D) 2 3 5 6 7 9

E) None of the above

14) When sorting an array of objects, if the values in the data member being sorted on are out of order for two objects, it is necessary to \_\_\_\_\_\_\_\_.

A) examine a different data member

B) swap these two data values

C) swap the two objects

D) swap one-by-one all data members in the two objects

E) stop the sort

15) We can measure the complexity of an algorithm that solves a computational problem by

determining the number of \_\_\_\_\_\_\_\_ for an input of size *n*.

A) output statements it has

B) times it loops

C) basic steps it requires

D) variables it uses

E) operations it performs

16) A \_\_\_\_\_\_\_\_ search uses a loop to sequentially step through an array.

A) binary

B) unary

C) linear

D) relative

E) bubble

17) The \_\_\_\_\_\_\_\_ search is adequate for searching through small arrays, but not through large ones.

A) binary

B) linear

C) blind

D) bubble

E) random

18) If a binary search is used to search for the number 4 in the 11-element array shown below, which value will the 4 be compared to first?

1 2 3 4 6 7 8 9 10 12 17

A) 1

B) 7

C) 8

D) 9

E) None of the above

19) When searching for a particular object in an array of objects, it is necessary to compare the \_\_\_\_\_\_\_\_ to the value in each examined object's \_\_\_\_\_\_\_\_ .

A) search key, private member data

B) key field, public member data

C) search key, public member data

D) search key, key field

E) key field, search key

20) To determine that a value is not present in an unordered array of 50 items, linear search must examine an **average** of \_\_\_\_\_\_\_\_ values.

A) 1

B) 6

C) 25

D) 50

E) 51

21) To locate a value that is in an ordered array of 50 items, linear search must examine **at most** \_\_\_\_\_\_\_\_ values.

A) 1

B) 6

C) 25

D) 50

E) 51

22) To locate a value in an ordered array of 50 items, binary search must examine **at most** \_\_\_\_\_\_\_\_ values.

A) 1

B) 6

C) 10

D) 25

E) 50

23) A \_\_\_\_\_\_\_\_ algorithm arranges data into some order.

A) sorting

B) searching

C) ordering

D) linear

E) binary

24) Sorted data can be ordered \_\_\_\_\_\_\_\_.

A) from lowest to highest value

B) from highest to lowest value

C) using a bubble sort algorithm

D) using a selection sort algorithm

E) in all of the above ways

25) When an array is sorted from highest to lowest, it is said to be in \_\_\_\_\_\_\_\_ order.

A) reverse

B) forward

C) descending

D) ascending

E) downward

26) A bubble sort is being used to arrange the following set of numbers in ascending order:

8 6 4 9 3 7

After the first pass of the sort is completed, what order will the data be in?

A) 3 4 6 7 8 9

B) 3 6 4 9 8 7

C) 6 4 8 3 7 9

D) 6 8 4 9 3 7

E) None of the above

27) A selection sort is being used to arrange the following set of numbers in ascending order:

8 6 4 9 3 7

After the first pass of the sort is completed, what order will the data be in?

A) 3 4 6 7 8 9

B) 3 6 4 9 8 7

C) 6 4 8 3 7 9

D) 6 8 4 9 3 7

E) None of the above

28) The \_\_\_\_\_\_\_\_ sort usually performs more exchanges than the \_\_\_\_\_\_\_\_ sort.

A) bubble, selection

B) selection, bubble

C) binary, linear

D) linear, binary

E) linear, bubble

29) Selection sort requires \_\_\_\_\_\_\_\_ passes to put **n** data items in order.

A) n

B) n/2

C) n/2 +1

D) n-1

E) n+1

30) We can estimate the \_\_\_\_\_\_\_\_ of an algorithm by counting the number of steps

it requires to solve a problem.

A) efficiency

B) number of lines of code

C) run time

D) code quality

E) result

**Part 2, True/False 是非題 (10%)**

**T**

1) Using a linear search, you are more likely to find an item than if you use a binary search.

F

2) When searching for an item in an **unordered** set of data, binary search can find the item more quickly than linear search.

T

3) When sorting an array of objects or structures, one must decide which data item to sort on.

T

4) Bubble sort and selection sort can also be used with STL vectors.

F

5) If algorithm A requires 2*n*+ 1basic operations to process an input of size *n*, and Algorithm B requires3*n + 2* basic operations to process the same input, algorithm A is considered to be more efficient than Algorithm B.

T

6) If algorithm A requires 2*n*+ 1basic operations to process an input of size *n*, and Algorithm B requires 3*n + 2* basic operations to process the same input, algorithms A and B are considered to be equally efficient.

T

7) A binary search requires that the elements be in order.

F

8) Using a binary search, you are more likely to find an item than if you use a linear search.

T

9) Any sorting algorithm, such as bubble sort or selection sort, that can be used on data stored in an array can also be used on data stored in a vector.

T

10) When sorting an array of objects, if the values in the data member being sorted on are out of order for two objects, those two data values should be swapped.

**Part 3, Coding 程式題(60%)**

1. **String Selection Sort**

Modify the selectionSort function presented in this chapter so it sorts an array of strings instead of an array of ints. Test the function with a driver program. Use the following skeleton to complete.

int main()

{

const int SIZE = 20;

string name[SIZE] =

{"Collins, Bill", "Smith, Bart", "Michalski, Joe", "Griffin, Jim",

"Sanchez, Manny", "Rubin, Sarah", "Taylor, Tyrone", "Johnson, Jill",

"Allison, Jeff", "Moreno, Juan", "Wolfe, Bill", "Whitman, Jean",

"Moretti, Bella", "Wu, Hong", "Patel, Renee", "Harrison, Rose",

"Smith, Cathy", "Conroy, Pat", "Kelly, Sean", "Holland, Beth"};

// Insert your code to complete this program.

}

1. **String Selection Sort Using files and vectors**

Modify the program you wrote for 1) so it reads in the 20 strings from a file. The data can be found in the **names.dat** file (you should use notepad to create such file). And it stores the names in a **vector of strings**, rather than in an array of strings. Create the vector without specifying a size and then use the **push\_back** member function to add an element holding each string to the vector as it is read in from a file. Instead of assuming there are always 20 strings, read in the strings and add them to the vector until there is no data left in the file.

1. **Binary String Search**

Modify the binarySearch function presented in this chapter so it searches an array of strings instead of an array of ints. Test the function with a driver program. Use previous program as a skeleton to complete. (The vector must be sorted before the binary search will work.)

1. **Sorting Benchmarks**

Write a program that uses two identical arrays of at least **30 integers** generated randomly. It should call a function that uses the **bubble sort** algorithm to sort one of the arrays in ascending order. The function should count the number of exchanges it makes. The program should then call a function that uses the **selection sort** algorithm to sort the other array. It should also count the number of exchanges it makes. Repeat the process 20 times and display these values of each time as well as the average values of exchanges for the two sorting methods on the screen.

1. **Search Benchmarks**

Write a program that has an array of at least **30 integers** generated randomly. It should call a function that uses the **linear search algorithm** to locate one of the values. All 30 of them must be searched. The function should keep a count of the number of comparisons it makes until it finds the value. The program then should call a function that uses the **binary search algorithm** to locate the same value in the original array. It should also keep count of the number of comparisons it makes. The function finally makes a copy of the array and sorts it, then uses the **binary search algorithm** to locate the same value in the sorted array. It should also keep count of the number of comparisons it makes. Display these values on the screen.

1. **Statistics Report of Monthly Rainfall**

Based on programming #4 in HW5, redesign your program so that in addition to the original function and report, it also displays the name (January,…) and year of each month in the period and its rainfall amount in, **sorted** in order of (1) **time** or (2) **rainfall** from highest to lowest or (3) **rainfall** from lowest to highest. You should modify your program accordingly. You should input at least **20 months of rainfall data** to illustrate your program. Make the program modular by calling on different functions to input the rainfall amounts, to sort the data, and to display the data.