University of Nevada, Las Vegas Computer Science 477/677 Spring 2022
Answers for Examination March 23, 2022

1. True or False. [5 points each]
   (a) T A binary search tree is a search structure.
   (b) T A minheap is a priority queue.
   (c) F A good programmer would never store data in an unordered list.

2. Fill in the blanks.
   (a) [10 points] Θ(n) What is the asymptotic complexity of merging two sorted lists, each of length n? Use Θ notation.
   (b) [10 points] A stack is a priority queue in which the most recently inserted item has priority.
   (c) [10 points] Dijkstra’s algorithm does not allow the weight of any arc to be negative.
   (d) [10 points] binary search is a divide and conquer algorithm which implements the operator find for an ordered list.
   (e) [10 points] fetch and store are operators of the ADT array.
   (f) [10 points] The items in a priority queue represent unfulfilled obligations.
   (g) [10 points] The worst case number of comparisons of any comparison/exchange sorting algorithm is Ω(n log n).
   (h) [10 points] radix sort is a sorting algorithm which does not use the comparison/exchange model of computation.
   (i) [20 points] quicksort and mergesort are divide-and-conquer sorting algorithms.
   (j) [20 points] What is the asymptotic time complexity for the Bellman-Ford algorithm on a weighted directed graph with n vertices and m edges, where, for some number p and for every vertex x, the least weight path from the source to x has no more than p edges? O(mp)
3. [20 points] Walk through Dijkstra’s algorithm for the following weighted directed graph.

![Graph Diagram]

4. [20 points] Write pseudocode for the Floyd–Warshall algorithm. We assume that $W[i,j]$ is the weight of the edge from $i$ to $k$, if there is one. If there isn’t one, we assume $W[i,j] = \infty$.

```
for all i and all j
{
    V[i,k] = W[i,j];
    back[i,j] = i;
}
for all i V[i,i] = 0;
for all j
    for all i and all k in either order
    {
        temp = V[i,j]+V[j,k];
        if (temp < V[i,k])
        {
            V[i,k] = temp;
            back[i,k] = back[j,k];
        }
    }
```
5. [20 points] What is the purpose of the function \texttt{george} below? Multiplication

Give a loop invariant for the main loop.

\[ \text{rslt } + ym = xn \]

```cpp
int george(int x, int n)
{
    // input condition: \( n \geq 0 \)
    int y = x;
    int m = n;
    int rslt = 0;
    while(m > 0)
    {
        if(m%2) // m is odd
        {
            m = m-1;
            rslt = rslt + y;
        }
        else
        {
            m = m/2;
            y = y+y;
        }
    cout << rslt;
}
}
```

6. Name each of these algorithms. 10 points each.

(a) \textbf{quicksort} Pick an element \( P \) from a set \( S \), then partition \( S \) into two parts: those items which are less than \( P \) and those greater than \( P \). Recursively sort each part, and combine them to form a sorted list.

(b) \textbf{mergesort} Divide a set \( S \) arbitrarily into two equal parts. Recursively sort each part, then combine the two sorted parts to obtain a sorting of \( S \).

(c) \textbf{binary search} Given a sorted set \( S \) and an item \( x \), you need to determine whether \( x \in S \). Pick one element, say \( m \), out of \( S \). If \( m = x \), you are done. If \( m < x \), discard \( m \) and all items of \( S \) which are greater than \( m \), while if \( x > m \), discard \( m \) all items which are all items of \( S \) which are less then \( m \). Keep doing this until you either find \( x \) or you have discarded all items of \( S \).

(d) \textbf{treesort} Given a set \( S \), create an empty binary search tree \( T \). Insert the items of \( S \) into \( T \) one at a time. Finally, visit and print the items of \( T \) in left-to-right order, also called inorder.

(e) \textbf{selection sort} Given a set \( S \), delete the least element of \( S \) and print it. Then delete the least remaining element of \( S \) and print it. Keep going until you have deleted and printed all elements of \( S \).

(f) \textbf{linear search} Look at each item in a list, starting at the head. If one of the items is equal to \( X \), then stop and report that you have found \( X \). If you reach the end of the list without finding \( X \), report that \( X \) is not in the list.
7. [20 points] Execute heapsort with input file ASQWFGKZ. Use the array below. Add additional rows if needed.
8. [20 points] The following code correctly computes the \( n^{\text{th}} \) Fibonacci number. However, it is not a good idea to use this code. Why not? How would you solve the same problem differently?

\begin{verbatim}
int fibonacci(int n)
{
    assert(n > 0)
    if(n <= 2) return 1;
    else return fibonacci(n-2) + fibonacci(n-1);
}
\end{verbatim}

It will take exponential time. Use dynamic programming or memoization instead.

9. [20 points] Find the strong components of the directed graph shown below, using the DFS method in our textbook.

![Diagram of a directed graph with nodes a, b, c, d, e, f, g, h, i, j, m and edges connecting them.]}