University of Nevada, Las Vegas Computer Science 477/677 Fall 2021

Assignment 7: Due Monday November 8, 2021 11:59 pm

Name:____

You are permitted to work in groups, get help from others, read books, and use the internet. Turn in the completed assignment on canvas, using instructions given to you by the grader, Mr. Heerdt, by 11:59 PM November 1.

In Problems XX–YY below we assume that G is a weighted directed graph with vertices numbered 0 through n-1. We let $E[i,j]$ be the weight of the arc from vertex i to vertex j. If no such arc exists, $E[i,j] = \infty$ by default.

1. Consider the single source shortest path problem on a weighted digraph G. (digraph = directed graph) Fill in each blank with **one** word, or formula. Assume G has n vertices and m arcs (directed edges.) Assume that every answer (shortest path from the source to a vertex) has no more than p edges.

   (a) The simple dynamic programming algorithm requires that G be ________________. In that case, the time complexity of the algorithm is ________________.

   (b) The Bellman-Ford algorithm requires that G have no ________________ ________________. In that case, the time complexity of the algorithm is ________________.

   (c) Dijkstra's algorithm requires that G have no ________________ ________________. In that case, the time complexity of the algorithm is ________________.

   (d) In order to work the simple dynamic programming algorithm for the single source shortest path problem, we must visit the vertices of G in ________________ order.

2. (a) Select a set of numbers from the list 2,6,3,1,8 which has the maximum total, but with the rule that you may not select any two consecutive numbers. (This is an instance of problem (b) below.)

   (b) Write, in pseudocode, an algorithm which writes a maximum total subsequence of a sequence of numbers, with the rule that no two consecutive members of the sequence may both be selected.

3. Explain how to implement a sparse array using a search structure. (This **exact** problem will be on the next examination on November 15.) Hint: You must explicitly describe how to implement the operators **fetch** and **store**.

4.
```
int f(int n)
{
 if(n<7) return 1;
 else return f(n/2)+f(n/2+1)+f(n/2+2)+f(n/2+3);
}
```

The function f(n) can be computed by recursion, as given in the C++ code above. However, we could also compute f(n) using dynamic programming or memoization.

(a) What is the asymptotic time complexity of the recursive computation of f(n)? (You should be able to solve this problem using one of the theorems we've covered, but if you can't, try programming it for various values of n.)

(b) What is the asymptotic time complexity of the dynamic programming computation of f(n)? (There is no excuse for not being able to figure this out without writing a program.)

(c) What is the asymptotic time complexity of the computation of $f(n)$ using memoization? (This is harder than the others. If you write a program and try various values of n, you need a range of large values, like 1024 and up, to get the picture. Remember: memoization uses a sparse array structure.)

(d) Read the handout johnson.pdf, which describes Johnson's algorithm. Work the exercise given on the last page.