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

Practice Examination for April 15, 2021

The entire practice examination is 310 points.

1. Fill in the blanks. [5 points each blank.]

   (a) An acyclic graph of 6 vertices has no more than ________ edges. (Exact answer.)

   (b) An acyclic digraph of 6 vertices has no more than ________ arcs. (Exact answer.)

   (c) A planar graph with 100 vertices has no more than ________________ edges. (Exact answer.)

   (d) An AVL tree of height 4 has at least ________ nodes. (Exact answer.)

2. Let G be a directed graph with n vertices and m arcs. Fill in the blanks, using asymptotic notation. [5 points each blank.]

   (a) The time complexity of the Floyd-Warshall algorithm is ________

   (b) The time complexity of Dijkstra's algorithm is ________, but there can be no ________________ arcs.

   (c) The time complexity of Johnson's algorithm is ________.

3. What is the asymptotic time complexity, in terms of n, of each of the following code fragments? (10 points each)

   (a)
```
for(int i = n; i > 0; i--)
  for(int j = i; j < n; j=2*j)
```

   (b)
```
for(int i = n; i > 0; i=i/2)
  for(int j = 1; j < i; j++)
```

4. Solve the recurrences. Give asymptotic answers in terms of n. Use O, Ω, or Θ, whichever is most appropriate. [10 points each.]

   (a) $F(n) \leq 2F(n/2) + 1$

   (b) $G(n) \geq 9G(n/3) + n^2$

   (c) $H(n) = 2H(\sqrt{n}) + \log n$

   (d) $K(n) = 4K(n/4) + 3K(n/2) + n$

5. The following C++ code computes a function f:

```
int f(int n) // input condition: n >= 0
 {
  if(n == 0) return 0;
  else return f(n/2)+f(n/3)+f(n/6) + n;
 }
```

(a) [10 points] What is the time complexity of the recursive code given above?

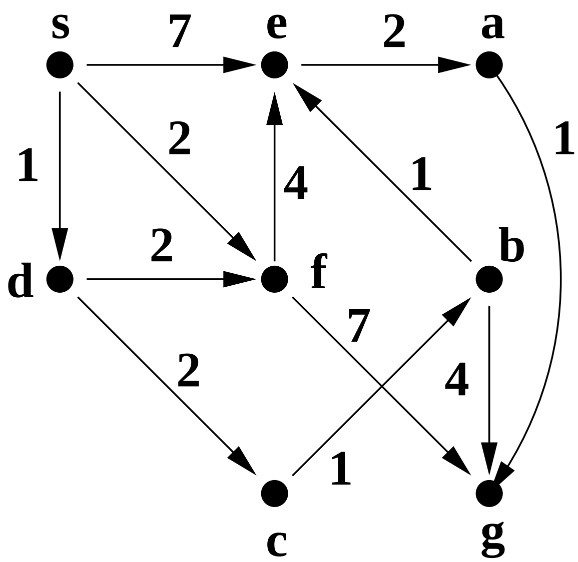
(b) [10 points] What is the time complexity of the dynamic programming computation of f(n)?

(c) [20 points] Write pseudocode for a computation of f(n) using memoization. (I won't ask you for the time complexity. The number of memos stored is $O(\log^2 n)$. If you count the time it takes to store and fetch memos, the time complexity is $O(\log^2 n \log\log n)$)

6. [20 points] What is the Hamming distance between 01101001 and 00101100?

7. [20 points] What is the Levenshtein distance between `abcdefg` and `acdgefb`? Show the matrix.

8. [20 points] Walk through Dijkstra's Algorithm for the following directed graph, where the start vertiex is s.



9. [20 points] Find the longest common subsequence of the sequences X = $\{1, 5, 9, 3, 6, 0, 4, 2, 7, 8\}$ and Y = $\{529403821\}$. Show the matrix.

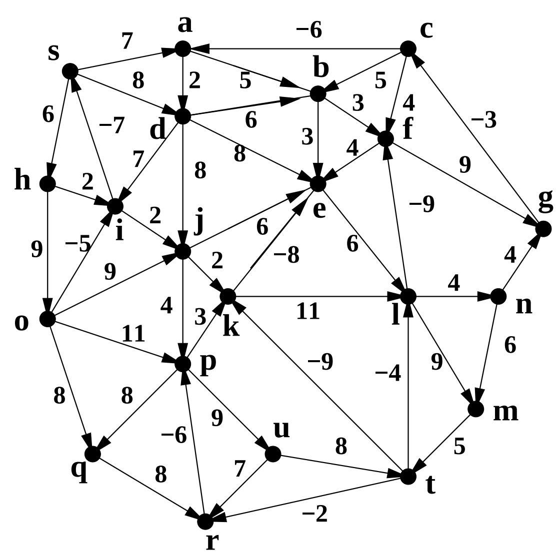
10. [20 points] Walk partway through Johnson's Algorithm for the all-pairs shortest path problem on the weighted digraph shown below.

   (a) Create a new graph by adding one node and n arcs.

   (b) Compute $h(v)$, the solution to the Bellman-Ford algorithm on each vertex. Note that $h(v) \leq 0$ for each vertex v. Indicate the value of $h(v)$ for each v.

   (c) Adjust the weights. No edge will have a negative weight. Indicate those adjusted weights.

   The next step is to use Dijkstra's algorithm to solve the single source shortest path problem n times, letting each vertex in turn be the source; but we will skip this step and all remaining steps.

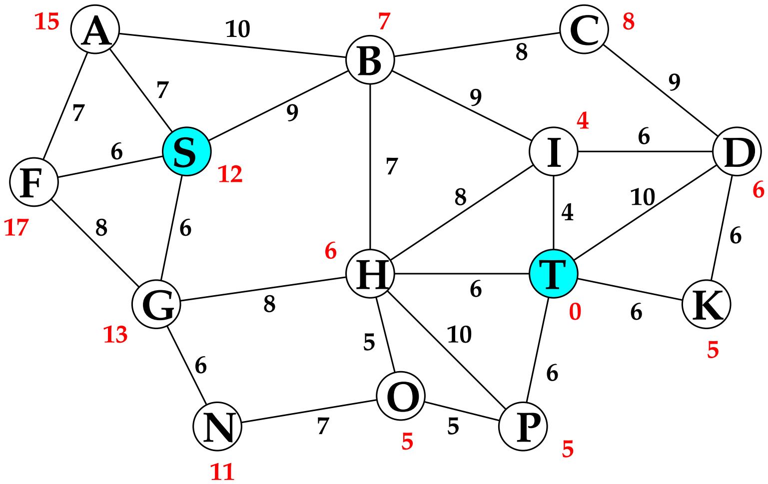


11. Each island in an ocean has some amount of treasure. A pirate ship is located at one island, and needs to reach its home island, picking up the treasure from each island it visits. However, due to prevailing winds, the ship can only sail South or East. The pirate captain wants to use Dynamic programming to chart the course, so as to maximize the treasure collected.

   (a) [10 points] What are the subproblems?

   (b) [10 points] What would be the arcs of the directed graph?

12. [20 points] Walk through the A* Algorithm for the weighted graph shown below. The heuristic is indicated by a red numeral at each vertex.



13. [20 points] You are given a sequence of n numbers, $x_1, \dots x_n$. A *range query* is a query $Q(i, j)$ for $i \leq j$, and must return the sum $x_i + \cdots x_j$. (Alternatively, it returns the maximum or minimum of the set $\{x_i, \dots x_j\}$.) If the sequence is stored in an array, a query $Q(i, j)$ takes $O(n)$ time to answer. Suppose we anticipate $O(n)$ queries. If we store the matrix of values of all $Q(i, j)$, Each query can be answered in $O(1)$ time, but it takes $O(n^2)$ time to initialize that matrix and $O(n^2)$ space to store it.

However, we can do better. There is a data structure which can be initialized in $O(n \log n)$ time and takes $O(n \log n)$ space and which then allows each query to be answered in $O(1)$ time.

We can do even better! There is a data structure which can be initialized in $O(n \log \log n)$ time and takes $O(n \log \log n)$ space and which still allows each query to be answered in $O(1)$ time.

Can we do even better than that?

You've encountered information theory before, although you may not have heard that name. The proof that no sorting algorithm on n items can use fewer than $\log_2(n!)$ comparisons in the worst case uses information theory.

14. You have a balance scale. When you put weights in both trays, the heavier weight side will go down, the other up. If the weights are the same, the two sides will stay at the same level.

    Suppose we have three coins, two of which are "good," they weigh exactly the same, and one "bad," meaning it is slightly too heavy. How can you determine, in one weighing, which coin is bad?

    Answer: put one coin in each tray. If they balance, the third coin is bad; otherwise, the bad coin is in the tray that goes down.

    (a) [10 points] Suppose we have nine coins, eight of which are "good," and weigh exactly the same, and one "bad," meaning it is slightly too heavy. How can you determine, in two weighings, which coin is bad?

    (b) [0 points] Suppose we have twelve coins, eleven of which are "good," and weigh exactly the same, and one "bad," meaning it is it is either too heavy or too light. How can you determine, in three weighings, which coin is bad and whether it is too heavy or too light? (You can find a solution on the internet, of course. But try it yourself.)

    (c) [0 points] Same problem as above, but we have thirteen coins. Use information theory to prove that there is no solution.