CECS 528, Exam 1, February 17th, Spring 2023, Dr. Ebert

Rules for Completing the Problems

NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION allowed when solving these problems. Make sure all these items are put away BEFORE looking at the problems. FAILURE TO ABIDE BY THESE RULES MAY RESULT IN A FINAL COURSE GRADE OF F.

Directions

Choose up to **six problems** to solve. Clearly mark each problem you want graded by placing an X or check mark in the appropriate box in the Grade(?) row of the table below. **If you don't mark any problems for us to grade or mark 7 or more problems, then we will record grades for the 6 that received the *fewest* points.**

Problem	1	2	3	4	5	6	LO1
Grade?							
Result							

Your Full Name:

Your Class ID:

1. For each of the following sentences, replace the lowercase roman numerals with the appropriate word or phrase. For the phrase "subject to the constraint", provide any conditions that must be met in order for the greedy choice to be valid. (6 points per algorithm)

   a. The greedy choice made in Kruskal's algorithm is to choose the **(i)** of least **(ii)** subject to the constraint that **(iii)**.

      (i)

      (ii)

      (iii)

   b. The greedy choice made in the `Fractional Knapsack` algorithm is to choose the **(i)** of greatest **(ii)** subject to the constraint that **(iii)**.

      (i)

      (ii)

      (iii)

   c. The greedy choice made in the `Unit Task Scheduling` algorithm is to choose the **(i)** of greatest **(ii)** subject to the constraint that **(iii)**.

      (i)

      (ii)

      (iii)

   d. The greedy choice made in the `Task Selection` algorithm is to choose the **(i)** of least **(ii)** subject to the constraint that **(iii)**.

      (i)

      (ii)

      (iii)

2. Suppose we have established that $\log^k n = o(n^\epsilon)$, for some integer $k > 0$ and $\epsilon > 0$. Use this fact to prove that $\log^{k+1} n = o(n^\epsilon)$. Note: correctly solvng this problem counts for passing LO1. (25 points)

3. Use the substitution method to prove that, if $T(n)$ satisfies

$$T(n) = 7T(n/4) + 2T(n/3) + n^2,$$

Then $T(n) = \mathrm{O}(n^2)$. Note: correctly solvng this problem counts for passing LO4. Hint: remember to state the inductive assumption. (25 points)

4. Suppose a function $T(n)$ satisfies

$$T(n) = 64T(n/b) + n^3,$$

for some integer $b \geq 2$. Based on this information, provide the tightest possible asymptotic lower and upper bounds for the growth of $T(n)$. Defend your answers. Hint: for example, $\Omega(n^2)$ is a tighter lower bound than $\Omega(n)$ since the former places more restriction on the possible growth of $T(n)$. Note: correctly solvng this problem counts for passing LO4. (25 pts)

5. Answer the following with regards to a correctness-proof outline for Dijkstra's algorithm. Note: correctly solvng this problem counts for passing LO3.

   a. Complete the following sentence. "In relation to Dijkstra's algorithm, an ***i*-neighboring path** from source s to a vertex v that is external to DDT_i is ..." (5 points)

   b. Complete the following sentence. "Furthermore, the ***i*-neighboring distance** $d_i(s, v)$ from source s to v is *defined as* ..." (5 points)

   c. Complete the following sentence. "The greedy choice made by Dijkstra's algorithm at round $i + 1$ is to select the external vertex v^* which has ..." (5 points)

   d. If P is any path from s to v^* (from part c), explain why $\text{cost}(P) \geq d_i(s, v^*)$. Conclude that $d(s, v^*) = d_i(s, v^*)$. (10 points)

6. For use in the `Unit Task Scheduling` algorithm, the `MNode` data structure may be defined as follows.

```
struct MNode
{
    MNode parent; //a reference to this node's parent
                  //assigned null in case this node is the root of its tree
    int index; //the scheduling-array index associated with this node
};
```

Provide a *recursive* implementation of the function

```
MNode root(MNode node);
```

which returns a reference to the root of the tree where node is located, *and* has the side-effect of performing path compression on the path from node to the root. Note: points will be lost if your implementation does not use recursion. (25 pts)

LO2. For the weighted graph with edges

$$(a, c, 5), (b, c, 4), (c, e, 6), (c, f, 3), (c, d, 2), (d, f, 1),$$

Show how the MTree forest of disjoint set data structure changes when processing each edge in Kruskal's sorted list of edges. When unioning two trees, use the convention that the root of the union is the root which has the *lower* alphabetical order. For example, if two trees, one with root a, the other with root b, are to be unioned, then the unioned tree should have root a. Your solution should include six snapshots, one for the processing of each edge. (0 pts)