Quiz 2

- Do not open this quiz booklet until directed to do so. Read all the instructions on this page.
- When the quiz begins, write your name on every page of this quiz booklet.
- You have 120 minutes to earn 100 points. Do not spend too much time on any one problem. Read them all through first, and attack them in the order that allows you to make the most progress.
- This quiz booklet contains 7 pages, including this one. Two extra sheets of scratch paper are attached. Please detach them before turning in your quiz at the end of the exam period.
- This quiz is closed book. You may use **two** $8\frac{1}{2}'' \times 11''$ or A4 crib sheet (both sides). No calculators or programmable devices are permitted. No cell phones or other communications devices are permitted.
- Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem. Pages may be separated for grading.
- Do not waste time and paper rederiving facts that we have studied. It is sufficient to cite known results.
- Show your work, as partial credit will be given. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it. Be neat.
- Good luck!

Problem	Parts	Points	Grade	Grader
1	14	15		
2	1	15		
3	1	15		
4	2	20		
5	1	15		
6	2	20		
Total		100		

Name: ____[1 point]

Friday Recitation:	Krzysztof **11 AM**	Alex **12 PM**	Zoran **3 PM**	Rishabh **4 PM**

Problem 1. True or False [14 points] (14 parts)

For each of the following questions, circle either T (True) or F (False).

(a) T F Heapsort is not a stable sorting algorithm.

(b) T F Finding the minimum element of a binary min heap takes logarithmic time.

(c) T F Any type of sorting algorithm can be used to sort the digits in one phase of radix sort.

(d) T F Given a matrix representation of a graph with V vertices, we can run depth-first search in O(V) time.

(e) T F DFS finds the longest paths from start vertex s to each vertex v in the graph.

(f) T F In a graph with negative weight cycles, one such cycle can be found in $O(VE)$ time where V is the number of vertices and E is the number of edges in the graph.

(g) T F One always obtains the shortest path to each vertex, i.e., one using the minimum number of edges, using breadth-first search.

(h) T F A directed graph with a negative weight cycle has a topological ordering.

(i) T F If one can reach every vertex from a start vertex in a directed graph, then the graph is strongly connected.

(j) T F Topological sort can be performed using one breadth-first search procedure on the graph.

(k) T F Finding the strongly connected components in a graph requires $\Omega(VE)$ time where V is the number of vertices and E is the number of edges.

(l) T F If one transforms edge weights $w(u, v)$ to $w'(u, v) = w(u, v) - \lambda(u) + \lambda(v)$ for arbitrary $\lambda()$, then Dijkstra on the modified graph will return shortest paths in the original graph.

(m) T F The only operations required by Dijkstra on the priority queue data structure are EXTRACT_MIN, INSERT and DECREASE_KEY.

(n) T F Dijkstra assumes the triangle inequality on edge weights, that is, for each triple of edges (u, v), (u, a) and (a, v), we have $w(u, v) >= w(u, a) + w(a, v)$.

Problem 2. Merging Multiple Lists [15 points]

You are given K sorted lists of numbers. Each list has length N/K, so the total length of all of the lists is N. Describe an algorithm using heaps to merge these K lists of N/K numbers into a single sorted list of N numbers. Your algorithm should run in $O(N \log K)$ time, although you will receive partial credit for less efficient algorithms.

Problem 3. Assigning Directions [15 points]

Consider a mixed unweighted graph $G = (V, E)$ with both directed and undirected edges. Assume that initially there are no cycles in G which use only directed edges. Give an algorithm to assign direction to each of the undirected edges so that the completely directed graph so obtained has no cycles. Analyze the asymptotic complexity of the algorithm.

Problem 4. Modified Dijkstra [20 points] (2 parts) Modify Dijsktra's algorithm to find, out of all shortest paths, the one with fewest number of edges from a start vertex s to all other vertices.

(a) [10 points] Propose an augmented data-structure for each node for solving this problem.

(b) [10 points] Modify the RELAX function with your augmentation.

```
RELAX(u,v,w)

   if (______________________________) // relax condition

         then d[v] = d[u] + w(u,v)

              ______________________________

              parent[v] = u
```

Problem 5. Road Network [15 points]

Consider a road network modelled as a weighted undirected graph G with positive edge weights where edges represent roads connecting cities in G. However some roads are known to be very rough, and while traversing from city s to t we never want to take a route that takes more than a single rough road. Assume a boolean attribute $r[e]$ for each edge e which indicates if e is rough or not. Give an efficient algorithm to compute the shortest distance between two cities s and t that doesn't traverse more than a single rough road. (Hint: Transform G and use a standard shortest path algorithm as a black-box.)

Problem 6. Dynamic Programming [20 points] (2 parts)

Describe an algorithm, using dynamic programming, to find the number of binary strings of length N which do not contain any two consecutive 1's. For example for N=2, the algorithm should return 3 as the possible binary strings are 00, 01 and 10. You will receive greater credit for a more efficient algorithm.

(a) [10 points] State the set of subproblems that you will use to solve this problem and the corresponding recurrence relation to compute the solution.

(b) [10 points] Write iterative (non-recursive) pseudo-code to compute the solution. Analyze the running time of your algorithm.

SCRATCH PAPER

SCRATCH PAPER