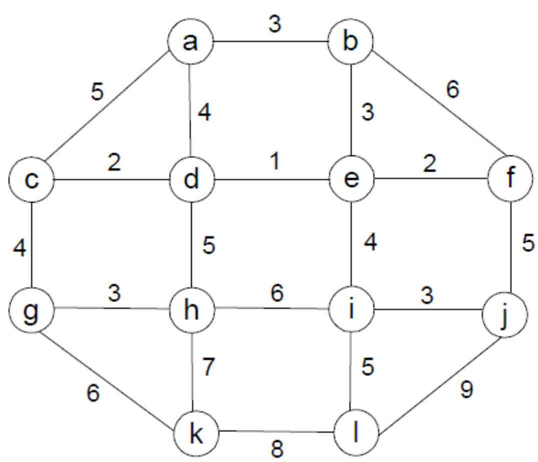
Analysis of Algorithms
CS 375, Spring 2019
Homework 16
Due **AT THE BEGINNING OF CLASS** Monday, April 29

- From your textbook (CLRS), please read Chapter 25.2, Chapter 16 up to (and including) Section 16.2, and Chapter 23.
- Some exercises on this problem set are adapted from exercises in our recommended Levitin textbook.
- When presenting an algorithm, describe it in English clearly, concisely, and unambiguously; pseudocode often helps clarify a presentation, but a pseudocode-only presentation is not acceptable. In general, unclear presentations may not receive full credit.
- *A general note:* When writing up your homework, please write neatly and **explain your answers clearly**, giving all details needed to make your answers easy to understand. Graders may not award credit to incomplete or illegible solutions. Clear communication *is* the point, on every assignment.

Exercises

1. Apply Kruskal's algorithm to the following graph, to find a minimum spanning tree. Show your work: To show that you understand Kruskal's algorithm, show the order in which each edge is added and, for each added edge, give a very brief (1 sentence or less is fine!) explanation of the reasons behind choosing that edge.



2. Consider the problem of scheduling n jobs of known durations $t_1, t_2, \ldots t_n$ for execution on a single-processor (single-core) computer. The jobs can be executed in any order, one job at a time. Ideally, the jobs would be scheduled in an order that minimizes the average time spent by all jobs in the system, where the time spent by a job in the system is the sum of the time spent waiting plus the time spent executing.

   For example, consider a case with two jobs j_1, j_2 with associated completion times $t_1 = 3, t_2 = 5$. If j_1 runs before j_2, then j_1 completes in time 3, j_2 completes in time 8 (3 for waiting plus 5 for executing), for a total time of 11 for all jobs and thus an average of 5.5. If instead j_2 runs before j_1, then j_2 completes in time 5, j_1 completes in time 8, and the total would be 13, for an average of 6.5.

(a) Design a greedy algorithm to schedule jobs, with the goal of minimizing the average completion time. Give a very brief explanation of what makes it a greedy algorithm.

(b) Does the greedy algorithm yield an optimal solution for every instance of this problem? If so, explain how we know it's always optimal; if not, give a concrete counterexample—here, a non-empty set of jobs and durations for which the algorithm is not optimal.

3. **Bridge Crossing Revisited!** Consider the generalization of the bridge crossing exercise earlier in the semester (HW1, exercise 5) in which there are $n > 1$ people whose bridge crossing times are $t_1, t_2, \ldots t_n$. All the other conditions of the problem are the same as before: at most two people at a time can cross the bridge (and they move with the speed of the slower of the two) and they must carry with them the only flashlight the group has.

   (a) Design a greedy algorithm for getting the entire group across the bridge, with the goal of minimizing the total time for the group to cross. Give a very brief explanation of what makes it a greedy algorithm.

   (b) Does the greedy algorithm yield an optimal solution (i.e., a minimal crossing time) for every instance of this problem? If so, explain how we know it's always optimal; if not, give a concrete counterexample **with the smallest number of people** for which it is not optimal.