Homework Assignment #2
Due: September 23, 2020, by 11:59 pm

- You must submit your assignment through the Crowdmark system. You will receive by email an invitation through which you can submit your work in the form of separate PDF documents with your answers to each question of the assignment. To work with a partner, you and your partner must form a group on Crowdmark. Crowdmark does not enforce a limit on the size of groups. The course policy that limits the size of each group to at most two remains in effect: submissions by groups of more than two persons will not be graded.
- It is your responsibility to ensure that the PDF files you submit are legible. To this end, I encourage you to learn and use the LaTeX typesetting system, which is designed to produce high-quality documents that contain mathematical notation. You are not required to produce the PDF files you submit using LaTeX; you may produce it any way you wish, as long as the resulting document is legible.
- By virtue of submitting this assignment you (and your partner, if you have one) acknowledge that you are aware of the policy on homework collaboration for this course.¹
- For any question, you may use data structures and algorithms previously described in class, or in prerequisites of this course, without describing them. You may also use any result that we covered in class, or is in the assigned sections of the official course textbooks, by referring to it.
- Unless we explicitly state otherwise, you may describe algorithms in high-level pseudocode or point-form English, whichever leads to a clearer and simpler description.
- Unless we explicitly state otherwise, you should justify your answers. Your paper will be graded based on the correctness and efficiency of your answers, and the clarity, precision, and conciseness of your presentation.

¹“In each homework assignment you may collaborate with at most one other student who is currently taking CSCC73. If you collaborate with another student on an assignment, you and your partner must submit only one copy of your solution, with both of your names. The solution will be graded in the usual way and both partners will receive the same mark. Collaboration involving more than two students is not allowed. For help with your homework you may consult only the instructor, TAs, your homework partner (if you have one), your textbook, and your class notes. You may not consult any other source.”

Question 1. (20 marks) The optimal length of a paddle for a canoeist is his/her shoulder height. We have $n$ canoeists with shoulder heights $h_1, h_2, \ldots, h_n$, and $n$ paddles of length $\ell_1, \ell_2, \ldots, \ell_n$. We must assign exactly one paddle to each canoeist (and, of course, different paddles to different canoeists). Such an assignment can be conveniently represented by a permutation $\pi$ of $1, 2, \ldots, n$: canoeist $i$ is assigned paddle $\pi(i)$. The penalty of such an assignment for canoeist $i$ is $|h_i - \ell_{\pi(i)}|$ — i.e., how far off the optimal is the length of the paddle assigned to the canoeist. The average penalty of such an assignment is $\frac{1}{n} \sum_{i=1}^{n} |h_i - \ell_{\pi(i)}|$. We wish to find an assignment that minimises the average penalty.

For each of the following two greedy strategies, either show that it determines a desired assignment or provide a counterexample:

a. Best-match first strategy: Find a pair $(i, j)$ that minimizes $|h_i - \ell_j|$ (i.e., a canoeist-paddle pair of minimum penalty). Assign paddle $j$ to canoeist $i$ (i.e., $\pi(i) = j$) and apply the same strategy to the remaining canoeists and paddles, until all canoeists are assigned paddles.

b. Increasing height strategy: Sort the canoeists in increasing shoulder height and the paddles in increasing length, and assign to each canoeist the corresponding paddle. In other words, the canoeist of
minimum shoulder height gets the shortest paddle, the canoeist of second minimum shoulder height gets the second shortest paddle, and so on.

**Question 2.** (10 marks) Let \( G = (V, E) \) be a directed graph, with non-negative weights assigned to the edges. The **bandwidth** of a path in \( G \) is the minimum weight of any edge on that path; the bandwidth of the (empty) path from a node to itself is defined to be infinite. (To understand this definition, think of the graph as a transportation network and the weight of an edge as the maximum amount of some product — data, vehicular traffic, oil — that can flow through the edge in this network. The maximum amount of the product that can flow through a path is limited by the most constrained edge on the path; this is the bandwidth of the path.) For a pair of vertices \( u \) and \( v \) in \( G \), the **maximum bandwidth** from \( u \) to \( v \) is the maximum bandwidth of any path from \( u \) to \( v \) (or zero, if no such path exists).

Modify Dijkstra’s algorithm so that it computes the maximum bandwidth from a given node \( s \) to all nodes. Your algorithm should have (asymptotically) the same running time as Dijkstra’s. Restate the claims that we used to prove the correctness of Dijkstra’s algorithm so that the new statements apply to your algorithm. (You need not prove the new statements, but they must be correct, and they must imply your algorithm’s correctness.)

**Question 3.** (10 marks) Consider Huffman’s algorithm.

a. Give an example of a (small) set of symbols and their associated frequencies so that the maximum frequency of any symbol is equal to \( 2/5 \), and Huffman’s algorithm could produce a tree in which no codeword has length 1. Explain why your example has the desired property.

b. Prove that for any set of symbols, if some symbol has frequency (strictly) greater than \( 2/5 \), Huffman’s algorithm will necessarily produce a codeword of length 1.