CIS 121 — Data Structures and Algorithms
Homework Assignment 7

Note: The homework is due electronically on Gradescope and Canvas on Monday, November 5 by 11:59 pm EDT. For late submissions, please refer to the Late Submission Policy on the course webpage. You may use a maximum of 2 late days on this homework.

A. Gradescope: You must select the appropriate pages on Gradescope. Gradescope makes this easy for you: before you submit, it asks you to associate pages with the homework questions. Failing to do so will get you 5% off, which cannot be argued against after the fact. Gradescope may prompt you with a warning to select your cover page, please ignore this warning and do not select the cover page.

B. Canvas: You must also submit your assignment on Canvas. Forgetting to do so will incur a 10% penalty.

C. \LaTeX: You must use the \texttt{hw121.cls} Latex template provided on the course website, or a 5% penalty will be incurred. Handwritten solutions or solutions not typeset in Latex will not be accepted.

D. Solutions: Please write concise and clear solutions; you will get only a partial credit for correct solutions that are either unnecessarily long or not clear. Please refer to the Written Homework Guidelines for all the requirements. Piazza will also contain a complete sample solution.

E. Algorithms: Whenever you present an algorithm, your answer must include 3 separate sections. Please see Piazza for an example complete solution.

1. A precise description of your algorithm in English. \textbf{No pseudocode, no code.}
2. Proof of correctness of your algorithm
3. Analysis of the running time complexity of your algorithm

F. Collaboration: You are allowed to discuss ideas for solving homework problems in groups of up to 3 people but \textit{you must write your solutions independently}. Also, you must write on your homework the names of the people with whom you discussed. For more on the collaboration policy, please see the course webpage.

G. Outside Resources: Finally, you are not allowed to use \textit{any} material outside of the class notes and the textbook. Any violation of this policy may seriously affect your grade in the class. If you’re unsure if something violates our policy, please ask.
1. [14 pts] Huffman Coding

a. [6] You are running the Huffman coding algorithm on a character set \( C = \{c_1, \ldots, c_n\} \) with \( n \geq 2 \) symbols whose frequencies are \( f_1, f_2, \ldots, f_n \). Suppose \( f_1 > (f_2 + f_3 + \ldots + f_n) \).

What must be the length of the codeword assigned to \( c_1 \) by the algorithm? Prove your answer.

b. [8] You got tired of problems that depend on variables like \( n \), so you decide to think about what would happen on smaller inputs. You come up with a character set with just 4 symbols, \( C = \{c_1, c_2, c_3, c_4\} \). Suppose that \( c_i \) occurs with frequency \( f_i \) for all \( 1 \leq i \leq 4 \), and that for this specific character set, \( f_1 \geq f_2 \geq f_3 \geq f_4 \). You find that the resulting encodings for each character is of length 2.

Find the maximum possible value of \( f_1 \). Prove your answer.

2. [19 pts] Well-Positioned Groups

Consider a group of \( n \) people standing in a line. We say the group is well-positioned with respect to a given parameter \( k \in [1..n] \) iff for any \( k \) consecutive people in the line, there are no two persons such that the height of one person is at least twice the height of the other person.

Design an \( O(n \log k) \) time algorithm that given the heights of \( n \) people standing in a line and the parameter \( k \), outputs whether or not they are well-positioned. You can assume that the heights are given in an array \( A[1..n] \) where \( A[i] \) is the height of the \( i \)-th person in the line; you should not make any assumptions about the range of the heights.

3. [10 pts] Graph Traversals

Consider the following weighted directed graph. Assume the adjacency list is in reverse sorted order based on the label of the vertices: for example, when iterating through the edges pointing from 0, consider the edge 0 → 6, then 0 → 3, and finally 0 → 1.

- **a. [2.5]** Run DFS on the graph above starting from vertex 0. List the vertices in order of their first visit.
b. [2.5] Run BFS on the graph above starting from vertex 0. List the vertices in order of their first visit.

c. [3] Run Dijkstra's algorithm on the graph above starting from vertex 0. Fill in the following chart; we’ve filled out the first couple entries for you. See the \LaTeX code on the last page for the chart — you must use this template. Also, please mark the vertex whose shortest distance was updated the most number of times with a ∗.

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Shortest Distance</th>
<th>Path from Vertex 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>—</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0 → 1</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>0 → 1 → 2</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

d. [2] True or false: this graph has a topological ordering. Give a one-sentence explanation for why or why not.

4. [19 pts] Roopa Takes to the Sea  

Wanting to pick up a side job outside of TAing, Roopa decides to try becoming a pirate. In particular, she has her sight set on a treasure located on the island of Fiji. However, in order to reach the island she must hop from one island to another since her electric pirate ship must be re-charged often. There are \( n \) total islands, where \( n \) is an even number, and \( m \) possible hops she can take between islands. Starting on the island of Tonga, all potential simple paths from Tonga to Fiji involve stopping at more than \( \frac{n}{2} \) islands. However, before she can begin her journey, Chris hears of her master plan and decides to try to stop her before she reaches Fiji.

a. Prove that there exists an island \( t \), which is different from Fiji and Tonga, such that if Chris can prevent Roopa from visiting this island, then he will block all possible paths from Tonga to Fiji.

b. Give an algorithm for finding such an island \( t \) in \( O(m + n) \) time.

5. [19 pts] Advance Registration  

Course registration opened, so you decide to go to Desirae for some advice on which CIS courses to take. Desirae has a list of all CIS courses \( C \), and another list \( P \) which lists all CIS courses and their associated pre-requisites, if any. She wants to see if you really learned the material in CIS 121, so she wants to quickly test your knowledge. To your surprise, Desirae learned some graph theory from her advisees!

Desirae gives you the graph \( G(C, P) \) whose edges \( P \) are a mix of directed and undirected edges. To be more precise, she gives you the list \( P = P_1 \cup P_2 \) where \( P_1 \) is the list of courses
with their associated pre-requisites (i.e., every edge in $P_1$ is an edge from a pre-requisite to a course), and $P_2$ is just a list of pairs of courses (i.e., undirected edges). You can assume that Desirae gives you $P_1$ and $P_2$ as two separate adjacency lists. She promises you that the listings in $P_1$ are correct, and that therefore you can safely assume no cycles exist in $G(C, P_1)$. Your task is to assign a direction to each of the edges in $P_2$ so that $G(C, P)$ remains acyclic. That is, Desirae wants you to tell her for each pair $(c_1, c_2) \in P_2$, which course could be a pre-requisite of the other, so that once everything is put together it all could make sense.

Desirae has no time to waste; she wants a linear time $O(|C| + |P|)$ time algorithm that assigns a direction to each of the edges in $P_2$ such that $G(C, P)$ remains acyclic.

6. [19 pts] Sarah & Co. Takes the Met Gala  
Now that Sarah & Co. has made a huge splash across the globe, demand for Sarah’s items has skyrocketed. Sarah was recently asked by Meghan Markle - the Duchess of Sussex - to dress her for the Met Gala. A huge fan of the royal family, Sarah took on the project straight away, and spent weeks designing and piecing together the perfect look for Meghan – she had designed a masterpiece!

After sending Meghan off the night of the Met Gala, Sarah plopped down in her store, completely exhausted from a long day’s work. While scrolling through her Instagram feed, Sarah noticed something glimmering from the corner of her eye. Her heart dropped – Meghan had forgotten her necklace! Sarah knew she had to figure out a way to get the necklace to Meghan as soon as possible – after all, the necklace was the centerpiece of the look that Sarah had designed!

Sarah quickly pulled up a map of Manhattan and began working to figure out the fastest way to get from her store to the Met. This map of Manhattan had $n$ buildings and $m$ two-way streets connecting some pairs of buildings. Luckily enough, the map also listed the amount of time it takes to get through each street, and oddly enough, no street takes longer than 15 minutes to travel through and all the times are positive integers. Sarah knew that she could simply use Dijkstra’s algorithm to find a path from her store to the Met (both of which were listed on her map as buildings), but she was sure it would take too long. Come up with a faster algorithm - one that runs in $O(n + m)$ time - to help Sarah determine the shortest amount of time it’ll take her to get to the Met. Sarah knows she can solve this by modifying the priority queue in Dijkstra’s algorithm, but she has modified enough data structures at this point, so she wants you to come up with an alternate solution.

**Feedback:**  How long did you spend on this assignment? What did you think of it, and how can we improve written assignments in the future? Let us know anonymously at: https://tinyurl.com/CIS121-18fa-hw-feedback

We really appreciate your feedback!
**\LaTeX** Code for Q3c’s chart: We will also include this in the post that releases the homework in case you are having difficulties copy pasting from here. Please check that post on Piazza.

\begin{center}
\begin{tabular}{|c|c|c|}
\hline
\textbf{Vertex} & \textbf{Shortest Distance} & \textbf{Path from Vertex 0} \\
\hline
0 & 0 & --- \\
1 & 3 & $0 \rightarrow 1$ \\
2 & 5 & $0 \rightarrow 1 \rightarrow 2$ \\
3 & & \\
4 & & \\
5 & & \\
6 & & \\
7 & & \\
8 & & \\
\hline
\end{tabular}
\end{center}