

- The assignment is due at Gradescope on Monday, November 14 at 11:59pm. Late assignments will not be accepted. Submit early and often.
- You are permitted to study with friends and discuss the problems; however, *you must write up your own solutions, in your own words*. Do not submit anything you cannot explain. If you do collaborate with any of the other students on any problem, please do list all your collaborators in your submission for each problem.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly prohibited.
- We require that all homework submissions are prepared in Latex. If you need to draw any diagrams, however, you may draw them with your hand. Please use a *new page to begin each answer*.

PROBLEM 1 *Another All-pairs short paths*

Let $G = (V, E)$ be a directed graph with edge weights $w(e)$ and no negative cycles.

1. Write one sentence that explains the variable $d(i, j, k)$ used in the Floyd-Warshall All-pairs shortest path algorithm discussed in class.
2. State the run time of the All-pairs shortest path algorithm discussed in class.
3. Consider the following algorithm.

NEWALLPAIRS(G, w)

- 1 Add a new node s' to G . Add edges of weight 0 from s' to every vertex $v \in V$. Call this new graph G' .
- 2 Run BELLMANFORD(G', s') to produce shortest path lengths $\delta(s', v)$.
- 3 For each $e = (x, y) \in E$, set $w'(e) \leftarrow w(e) + \delta(s', x) - \delta(s', y)$
- 4 For each $v \in V$, run DIJKSTRA(G, v, w') to compute $\delta(v, w)$ for all $w \in V$.
- 5 Set $d_{v,w} \leftarrow \delta(v, w) - \delta(s', v) + \delta(s', w)$

We aim to analyze why this algorithm works. The first step is to argue that the new edge weights w' that are defined in step (3) are all non-negative.

Prove that for all $e \in E$, $w'(e) \geq 0$.

4. This explains why we can use the fast DIJKSTRA algorithm with weight w' in step (4) to compute shortest paths from node $v \in V$ to all other nodes in the graph. However, we must argue that the shortest paths under w' and under w will be the same shortest path.

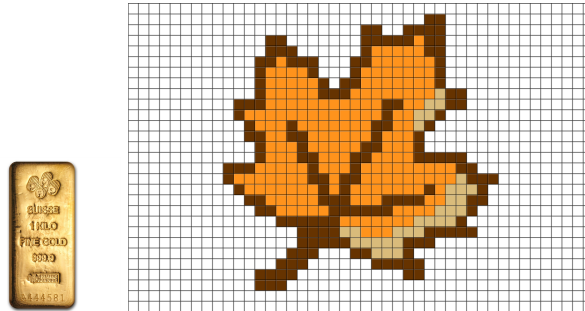
Prove that for any pairs of nodes $u, v \in V$, if p is a shortest path from u to v with respect to weight function w' , then p is also a shortest path from u to v with respect to weight function w .

5. What is the running time of NEWALLPAIRS in terms of V and E ? When does this algorithm run faster than the All-pairs algorithm discussed in class?

PROBLEM 2 *Gold Bullion face*

In the ruins of Pompeii, I remember seeing the [House of the Tragic Poet](#) with a famous mosaic floor proclaiming visitors to “Beware of the Dog.” In Boston, a less tragic and wealthier poet has commissioned a mosaic using 1kg bars of solid gold, specifically the type CreditSuisse mints in the dimension 80mmx40mm.

Design an algorithm that takes as input a grid of 40mmx40mm squares that are either colored or white. The algorithm determines if the colored squares in the design can be *entirely covered* with gold bullion bars. Note that gold bars can never be split in half! Each gold bar covers exactly two of the squares. As an example, consider the gold bars on the left, and the pixel art on the right. Can the leaf be covered in gold?



(Hint: formulate the question as a type of bipartite matching problem.)

PROBLEM 3 *Matrix rounding*

Suppose we are given a large matrix $A[1 \dots m][1 \dots n]$ of population data (each entry is a non-negative real number). We want to publish matrix A , but need to simplify it for the public by making the entries integers by replacing each entry x in A with either $\lceil x \rceil$ or $\lfloor x \rfloor$. However, the matrix represents important population data, and we do not want to change the sums of entries in any row or column. For example:

$$\begin{bmatrix} 1.2 & 3.4 & 2.4 \\ 3.9 & 4.0 & 2.1 \\ 7.9 & 1.6 & 0.5 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 4 & 2 \\ 4 & 4 & 2 \\ 8 & 1 & 1 \end{bmatrix}$$

Using max flow, describe and analyze an efficient algorithm that either rounds A in this fashion, or reports correctly that no such rounding is possible.