CSE548, AMS542: Analysis of Algorithms, Fall 2017 **Date:** Oct 6

Homework #1

(Due: Oct 20)

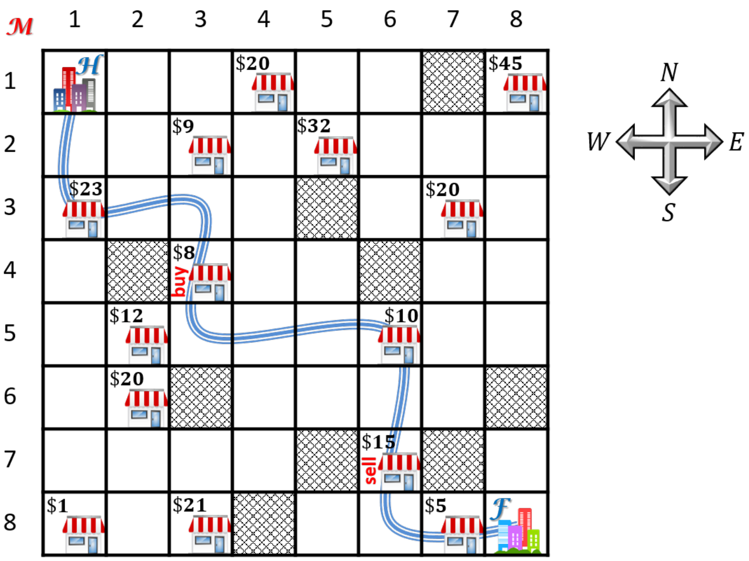

Figure 1: The $\mathcal{H}$ to $\mathcal{F}$ trade route shown in the grid map leads to a profit of $7/item. No other route is more profitable.

Task 1. [60 Points] Most Profitable Trade Routes

Trader $\mathcal{T}$ specializes in the buying and selling of one specific type of item $\mathcal{I}$. He often undertakes long trips from his homeland $\mathcal{H}$ to a distant foreign land $\mathcal{F}$ buying and selling $\mathcal{I}$ at the trading posts along the route in order to make a good profit. He starts from $\mathcal{H}$ without any $\mathcal{I}$, buys $\mathcal{I}$s exactly once and sells them exactly once (can buy and sell at the same or different trading posts), and reaches $\mathcal{F}$ without any $\mathcal{I}$.

$\mathcal{T}$ has a grid map showing the locations of $\mathcal{H}$ (at the north-west corner), $\mathcal{F}$ (at the south-east corner) and all trading posts that buy/sell $\mathcal{I}$. At each trading post, $\mathcal{I}$ is bought and sold at the same price and that price is also shown on the map. Neither $\mathcal{H}$ nor $\mathcal{F}$ has a trading post that buys/sells $\mathcal{I}$s. During his trip, $\mathcal{T}$ always makes sure that he either moves to the east or to the south from his current location and thus he gets closer to $\mathcal{F}$ with every step. Some cells in the grid map are marked as inaccessible (because of harsh geography, hostile/conflict regions, etc.) and $\mathcal{T}$ must avoid them. Though $\mathcal{H}$ will never be marked as inaccessible, the same is not necessarily true for $\mathcal{F}$. If $\mathcal{F}$ is marked as inaccessible then there will be no trade route from $\mathcal{H}$ to $\mathcal{F}$.

Before he starts his trip, $\mathcal{T}$ uses the information given on the grid map to find a trade route that will allow him to make the biggest profit.

Figure 1 shows an example on an 8×8 grid map. A trade route is shown from $\mathcal{H}$ to $\mathcal{F}$ that results in a profit of \$7/item provided $\mathcal{T}$ buys $\mathcal{I}$s for \$8/item from the trading post in cell $(4, 3)$ and sells all of them for \$15/item at the trading post in cell $(7, 6)$. The path can be specified using the string "*SSEESSEEESSSEE*" where an '*S*' means that $\mathcal{T}$ moves to the cell to the south of the current cell and an '*E*' means that he moves to the cell to the east of the current cell. No other trade route leads to a higher profit per item.

In this task, you will solve the following problem (denoted as problem $\mathcal{M}$). Given an $m \times n$ grid map $\mathcal{M}$ as described above (with $mn > 2$), find a most profitable trade route from $\mathcal{H}$ (at cell $(1, 1)$) to $\mathcal{F}$ (at cell (m, n)) if such a route exists, otherwise report that none exists. The route must be described using a sequence of '*S*'s and '*E*'s as explained in the previous paragraph.

Now answer the following questions.

(a) [**10 Points**] Suppose $\mathcal{M}_L$ denotes the sub grid map consisting of the leftmost $n-1$ columns of $\mathcal{M}$ and $\mathcal{M}_T$ denotes the one consisting of the topmost $m-1$ rows of $\mathcal{M}$. Show that if you know the solutions to both $\mathcal{M}_L$ and $\mathcal{M}_T$ you can extend the solution to $\mathcal{M}$ in $\Theta(1)$ time.

(b) [**5 Points**] Explain how you will use the observation from part (a) to solve problem $\mathcal{M}$ in $\Theta(mn)$ time and $\Theta(mn)$ extra space.

(c) [**35 Points**] Suppose $m = n = 2^k$ for some integer $k > 0$. Design a recursive divide-and-conquer algorithm to solve problem $\mathcal{M}$ in $\Theta\left(n^2\right)$ time and $\Theta(n)$ space. Find and solve the recurrences for running time and space usage.

Task 2. [50 Points] Gravity

For simplicity, in this task we will assume that we live in a flat 2D universe[1]. We will pick a large square region $\mathcal{S}$ of that universe and for every point $p \in \mathcal{S}$, compute the net gravitational force exerted by the celestial objects in $\mathcal{S}$ on an object of unit mass placed at p. But in order to make sure that we can solve this problem using the finite amount of computational resources available to us, we overlay an $n \times n$ square grid on $\mathcal{S}$ and only compute the net gravitational force acting at the center of each grid cell due to objects in other cells. Let's assume for simplicity that no celestial object spans more than one grid cell. We assume that $\mathcal{S}$ contains up to $\Theta\left(n^2\right)$ celestial objects and each such object is mapped to the center of the grid cell it belongs to. However, a grid cell may contain multiple objects, and if k objects of masses $m_1, m_2, \ldots, m_k$ belong to a cell then we will assume that the cell contains only one object of mass $\sum_{i=1}^{k} m_i$. Thus we can assume that each grid cell has exactly one mass m at its center. If $m = 0$ then the cell does not contain any objects, otherwise m is the sum of the masses of all objects in that cell.

We will use Newton's formula to compute gravitational forces assuming that all masses are point masses. If x and y are two objects of mass m_x and m_y, respectively, and r is the distance between their centers of mass, then the gravitational force $\overrightarrow{F}$ exerted by x on y will be in the direction from y to x and have value:

$$|\overrightarrow{F}| = G\left(\frac{m_x m_y}{r^2}\right),$$

[1] extension to 3D is straightforward but cumbersome

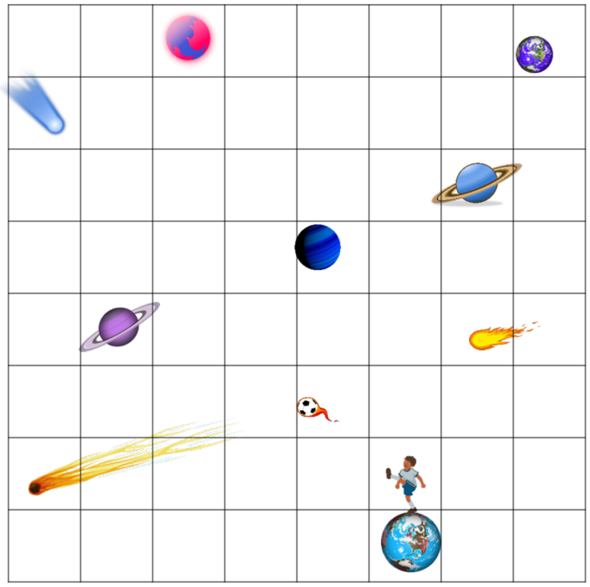

Figure 2: A square-shaped region of our flat 2D universe.

where G is the gravitational constant.

When summing up multiple gravitional forces acting on a single object one must take the directions also into account, that is, perform vector addition and not scalar addition.

Now answer the following questions.

(a) [**10 Points**] Give an algorithm that computes the net gravitional forces acting on unit-mass objects at all cells of the $n \times n$ grid in $\mathcal{O}\left(n^4\right)$ time.

(b) [**40 Points**] Explain how you will solve the problem given in part (a) in $\mathcal{O}\left(n^2 \log n\right)$ time.

Task 3. [70 Points] Rotator Matrices

We will call an $m \times m$ square matrix $\mathcal{M}$ a *rotator matrix* provided for each $i \in [2, m]$, its i^{th} row can be obtained simply by rotating the entries of its first row and no two rows are the same. Figure 3(a) shows an example of a 5×5 rotator matrix in which rows 2, 3, 4 and 5 are obtained by rotating row 1 to the left by 2, 4, 1 and 3 positions, respectively.

Now suppose n and k are positive integers such that $n^{\frac{1}{k}}$ is also an integer. Let $\mathcal{B}$ be an $n \times n$ matrix composed of $\frac{n}{n^{\frac{1}{k}}} \times \frac{n}{n^{\frac{1}{k}}} = \frac{n^2}{n^{\frac{2}{k}}}$ nonoverlapping rotator submatrices of size $n^{\frac{1}{k}} \times n^{\frac{1}{k}}$ each. Figure 3(b) shows an example for $n = 9$ and $k = 2$.

Given an arbitrary $n \times n$ matrix $\mathcal{A}$ this task asks you to multiply $\mathcal{A}$ by $\mathcal{B}$, that is, compute $\mathcal{A} \times \mathcal{B}$. You can assume that all entries of both matrices are real numbers.

(a) [**40 Points**] Give an algorithm that can compute $\mathcal{A} \times \mathcal{B}$ in $o\left(n^{2.41}\right)$ time when $k = 2$. You must clearly prove that the time complexity of your algorithm is, indeed, $o\left(n^{2.41}\right)$.

$$\begin{bmatrix} 5 & 2 & 4 & 1 & 9 \\ 4 & 1 & 9 & 5 & 2 \\ 9 & 5 & 2 & 4 & 1 \\ 2 & 4 & 1 & 9 & 5 \\ 1 & 9 & 5 & 2 & 4 \end{bmatrix}$$

(*a*) An example 5 × 5 rotator matrix.

$$\left[\begin{array}{ccc|ccc|ccc} 7 & 3 & 4 & 6 & 1 & 9 & 4 & 3 & 4 \\ 4 & 7 & 3 & 1 & 9 & 6 & 3 & 4 & 4 \\ 3 & 4 & 7 & 9 & 6 & 1 & 4 & 4 & 3 \\ \hline 0 & 5 & 7 & 1 & 2 & 3 & 5 & 7 & 0 \\ 7 & 0 & 5 & 2 & 3 & 1 & 0 & 5 & 7 \\ 5 & 7 & 0 & 3 & 1 & 2 & 7 & 0 & 5 \\ \hline 8 & 6 & 4 & 0 & 0 & 1 & 2 & 8 & 9 \\ 6 & 4 & 8 & 1 & 0 & 0 & 9 & 2 & 8 \\ 4 & 8 & 6 & 0 & 1 & 0 & 8 & 9 & 2 \end{array}\right]$$

(*b*) A 9 × 9 matrix composed of 9 nonoverlapping 3 × 3 rotator submatrices.

Figure 3: A rotator matrix and a matrix composed of smaller rotator submatrices.

(*b*) [**30 Points**] What if $k \neq 2$? Extend your algorithm and derive its time complexity for general (positive integral values of) k.