Due: In tutorial on March 17 by 12:10pm

1. Consider the problem of multiplying two $n \times n$ matrices of integers, computing $\mathbf{C} = \mathbf{AB}$. The classical method simply computes

$$c_{i,j} = \sum_{k=1}^{n} a_{i,k}\, b_{k,j}$$

and runs in $\Theta(n^3)$ time.

Strassen's method improves the asymptotic running time to $\Theta(n^{\log_2 7})$. When n is a power of 2, we perform the following steps:

- Divide each matrix into four $\frac{n}{2} \times \frac{n}{2}$ submatrices:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathbf{C}_{1,1} & \mathbf{C}_{1,2} \\ \mathbf{C}_{2,1} & \mathbf{C}_{2,2} \end{bmatrix}$$

- Compute the following products recursively:

$$\begin{aligned}
\mathbf{M}_1 &= (\mathbf{A}_{1,1} + \mathbf{A}_{2,2})(\mathbf{B}_{1,1} + \mathbf{B}_{2,2}) \\
\mathbf{M}_2 &= (\mathbf{A}_{2,1} + \mathbf{A}_{2,2})\mathbf{B}_{1,1} \\
\mathbf{M}_3 &= \mathbf{A}_{1,1}(\mathbf{B}_{1,2} - \mathbf{B}_{2,2}) \\
\mathbf{M}_4 &= \mathbf{A}_{2,2}(\mathbf{B}_{2,1} - \mathbf{B}_{1,1}) \\
\mathbf{M}_5 &= (\mathbf{A}_{1,1} + \mathbf{A}_{1,2})\mathbf{B}_{2,2} \\
\mathbf{M}_6 &= (\mathbf{A}_{2,1} - \mathbf{A}_{1,1})(\mathbf{B}_{1,1} + \mathbf{B}_{1,2}) \\
\mathbf{M}_7 &= (\mathbf{A}_{1,2} - \mathbf{A}_{2,2})(\mathbf{B}_{2,1} + \mathbf{B}_{2,2})
\end{aligned}$$

- Compute the answer by the following terms:

$$\begin{aligned}
\mathbf{C}_{1,1} &= \mathbf{M}_1 + \mathbf{M}_4 - \mathbf{M}_5 + \mathbf{M}_7 \\
\mathbf{C}_{1,2} &= \mathbf{M}_3 + \mathbf{M}_5 \\
\mathbf{C}_{2,1} &= \mathbf{M}_2 + \mathbf{M}_4 \\
\mathbf{C}_{2,2} &= \mathbf{M}_1 - \mathbf{M}_2 + \mathbf{M}_3 + \mathbf{M}_6
\end{aligned}$$

Though Strassen's method has slower asymptotic growth, the hidden constant is large.

(a) Give closed form expressions for the *exact* number of integer multiplications and for the *exact* number of integer additions performed by the classical method when n is a power of 2. Show your work.

(b) Give closed form expressions for the *exact* number of integer multiplications and for the *exact* number of integer additions performed by Strassen's method when n is a power of 2. (Consider integer subtraction equivalent to an addition.) Show your work.

(c) Suppose integer multiplication is m times more expensive than integer addition. We want to create a more efficient *hybrid algorithm* for matrix multiplication that performs Strassen's method for large matrices, but reverts to the classical method for small matrices. Determine the *crossover point*, where our hybrid algorithm should switch to the classical method, when $m = 10$, 5 and 1.1. Assume n is a power of 2, and show your work.

2. Given a set of points in the plane, $p_1 = (x_1, y_1), \ldots, p_n = (x_n, y_n)$, we want a list of all pairs of points closer than 2δ to each other, where δ is the minimum distance between any two points (*i.e.*, a list of all pairs (p_i, p_j) such that $d(p_i, p_j) < 2\delta$).

   Give an $\mathcal{O}(n \log n)$ algorithm that solves this problem. Justify that your algorithm is correct and runs within the required time bound (you may use the Master Theorem as long as you explain clearly how to apply it to your recurrence and you justify briefly that your recurrence is correct).

3. Let $N(V, E)$ be a flow network and let f and f' be two different maximum flows found by the Ford-Fulkerson Network Flow Algorithm. The proof of the Ford-Fulkerson Theorem shows how to produce a minimum capacity cut from a maximum flow found by the algorithm.

   Decide whether you think the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.

   > *The minimum capacity cut that corresponds to f is the same minimum capacity cut that corresponds to f'.*

4. Consider the following natural disaster scenario. Due to large-scale flooding in a region, paramedics have identified a set of n injured people distributed across the region, who need to be rushed to hospitals. There are k hospitals and each of the injured people has to be taken to a hospital that is within a half-hour's driving time of their current location. (Note that some people will have options for which hospital they are taken to.)

   The paramedics want to make sure that in choosing a hospital for each injured person, they make sure that the load on the hospitals is *balanced* (i.e. each hospital receives at most $\lceil \frac{n}{k} \rceil$ people).

   Give a polynomial-time algorithm that takes the given information about the people's locations and determines whether this is possible. Justify briefly that your algorithm is correct and runs in polynomial time.

   [This question is based on exercise number 9 on page 419 of the textbook.]