# Extra/Makeup Programming Questions
# CS6515 (Spring 2025)

- The questions on this set can only be solved via programming.
- This problem set is due on **Friday May 2nd** (Friday after the final exam).
- Submission is via Gradescope.
- There are no bonus points.

The following are programming questions. If you solve them, they'll be graded as a 20/20 homework problem. You can use this problem set to makeup some previous homework problems. Since there are no bonus points that have to be verified by a TA and the submissions are autograded (with only quick verification by TA), we can afford to have a late due date. You can submit any time until May 2nd (Friday after the final exam). While there are no office hours after last day of class, you can still post questions on Piazza.

**Grades:** Gradescope will automatically test your submission. You can resubmit as often as you want (until the due date) and you can see on gradescope how many test cases you currently passed. There are no partial points for passing part of the test cases, because even very wrong solutions can pass some test cases. While your submission must pass all test cases on gradescope, the final pass/fail decision is made by a TA. (This is to make sure your submission is actually solving the problem described above, in case of buggy autograder, and to prevent "cheating" by hardcoding the output.)

## 30 Reduction: OV to $ST$-Diameter

**$ST$-Diameter** In the $ST$-Diameter problem, we are given an undirected unweighted graph $G = (V, E)$ and two sets $S \subset V$, $T \subset V$, and an integer $k$. We must decide if there exists a pair $s \in S, t \in T$ with $\text{dist}(s, t) > k$.

Remark: The naïve way to solve this problem would be to run Dijkstra from every $s \in S$ to compute the distances, and then check for each $t \in T$ if $\text{dist}(s, t) > k$. In worst-case, this could take up to $O(|V||E|)$ time when $S, T$ are large. In this question, we want to check if there could be a faster algorithm with, e.g., $O(|V|^{0.999}|E|)$ complexity or less.

**Problem:** Assuming there is a $T(|V|, |E|)$-time algorithm for $ST$-diameter, show there is a $O(T(n, nd))$-time algorithm for the OV-problem ($n$ = number of vectors, $d$=their dimension).

**Remark:** So if there was some fast $O(|V|^{0.999}|E|)$-time algorithm for $ST$-diameter, then there would be an $O(n^{1.999}d)$-time algorithm for OV, which contradicts the OV-conjecture.

**Hint:** Do not do any $O(n^2)$-time nested for-loops. That will be too slow. In particular, do not search for orthogonal vectors yourself. You want to find a nice graph with $O(n + d)$ vertices and $O(nd)$ edges (ie each vector is some vertex and has $O(d)$ edges. Though there could also be $O(d)$ additional vertices.). The graph should have the property that two vertices $u \in S, v \in T$ have a longer distance when the corresponding vectors are orthogonal.

It may help to think about the vectors $u_1, ..., u_n$ and $w_1, ..., w_n$ as two matrices $\mathbf{U}, \mathbf{W}$. How did we previously connect matrices with graphs?

### 30.1 Input/Output Format for Submission

**Input** There are code templates on Piazza that handle reading the input. The input will be of the following format. The first line is $n$ and $d$, separated by space. The next $2n$ lines are each a 0/1-string of length d. Each of the first $n$ lines represents a vector $u_1, ..., u_n \in \{0, 1\}^d$, and each of the next $n$ lines represents a vector $w_1, ..., w_n \in \{0, 1\}^d$. Example input:

```
2 2
10
01
11
01
```

This represents $u_1 = (1, 0), u_2 = (0, 1)$ and $w_1 = (1, 1), w_2 = (0, 1)$.

**Output** Your submission must print an input for the $ST$-diameter problem. so it must print an undirected graph $G = (V, E)$, sets $S \subset V$, $T \subset V$, and an integer $k$.
The first line printed by your submission must be 4 integers $|V|, |S|, |T|, k$. Then followed by $|S|$ lines, listing the vertices in $S$. Then $|T|$ lines listing the vertices in $T$. At last, print one line for each edge in $E$.
Here is some example input to the $ST$-diameter problem:

```
5 2 2 1
0
1
3
4
0 1
1 2
3 4
1 3
```

This is a graph with 5 vertices (0,1,2,3,4), set $S = \{0, 1\}$, $T = \{3, 4\}$, integer $k = 1$, and edges $\{0, 1\}$, $\{1, 2\}$, $\{3, 4\}$, $\{1, 3\}$. It asks if there is a pair $s \in S$, $t \in T$ with distance $> k$. (There is such a pair: $0 \in S, 4 \in T$ with $\text{dist}(0, 4) = 3 > k$ via path $0 \to 1 \to 3 \to 4$.)

Basically, after reading the input, your algorithm should solve the OV-problem via some ST-diameter algorithm. The call to "`STDiameter(G,S,T,k)`" is done by the autograder. So your program only needs to construct the graph $G$, sets $S, T$ and integer $k$.

There is no example output, since figuring out what the graph/sets/$k$ should look like is part of the exercise.

The test cases are public on Piazza. The provided output files only contain a 0 or 1 depending on whether there is an orthogonal pair. It is not an example output for your algorithm!

# 31 Randomized Algorithm

We are given 3 matrices $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{R}^{n \times n}$. It is claimed that $\mathbf{C} = \mathbf{AB}$. We want to verify whether that is true. The naive way would be to compute the matrix product $\mathbf{AB}$ and check whether each entry is the same as $\mathbf{C}$. However, this would be slow an take $O(n^3)$ time to multiply the matrices. We want to verify in only $O(n^2)$ if the matrix $\mathbf{C}$ is correct.

**Problem:** Write an $O(n^2)$-time algorithm that, given $\mathbf{A}, \mathbf{B}, \mathbf{C}$, returns true/false whether $\mathbf{C} = \mathbf{AB}$. Your algorithm is allowed to be randomized, but should be correct with high probability.

**Hint:** A matrix represents a linear map/function, i.e., a matrix $\mathbf{M}$ maps vector $v$ to $\mathbf{M}v$. So testing if $\mathbf{C}$ and $\mathbf{AB}$ are identical, is asking whether the are the same linear map/function. We talked in class (Thu 04/10) about how to test that certain functions are identical.

## 31.1 Input/Output Format for Submission

There is a template on Piazza for handling input/output. Test caes are also provided.

**Input/Output:** One line, specifying $n$. Then $3n$ lines, each consisting of $n$ space separated numbers. These lines specify $n \times n$ matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$. Example:

```
3
3 3 1
-3 3 2
-3 -4 0
-1 -1 0
2 -2 1
-1 -2 2
2 -11 5
7 -7 7
-5 11 -3
```

Here the matrices are of size $3 \times 3$ and we have

$$\mathbf{A} = \begin{bmatrix} 3 & 3 & 1 \\ -3 & 3 & 2 \\ -3 & -4 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -1 & -1 & 0 \\ 2 & -2 & 1 \\ -1 & -2 & 2 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 2 & -11 & 5 \\ 7 & -7 & 7 \\ -5 & 11 & -3 \end{bmatrix}$$

Your algorithm must print `True` or `False`, depending on whether we have $\mathbf{C} = \mathbf{AB}$. In the above example, the output should be `False` because entry $\mathbf{C}_{33} = -3$ but $(\mathbf{AB})_{3,3} = -3 \cdot 0 - 4 \cdot 1 + 0 \cdot 2 = -4$.