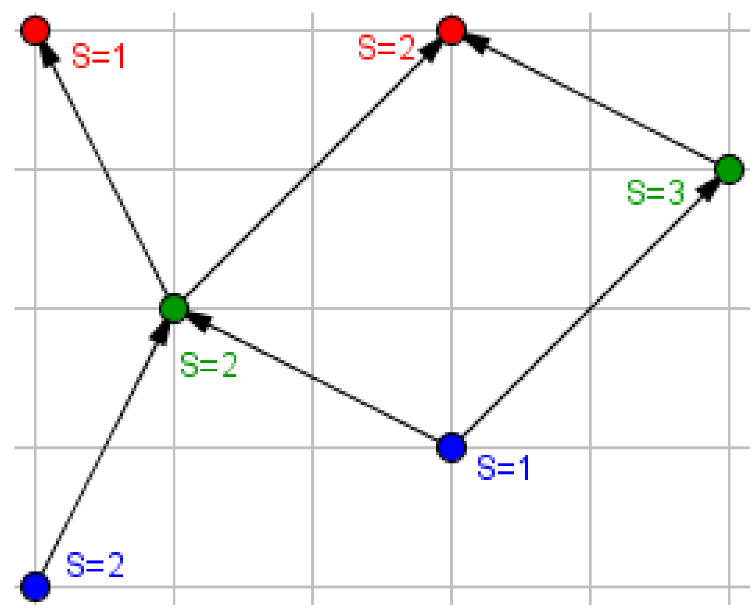
Task AT1: Walltopia

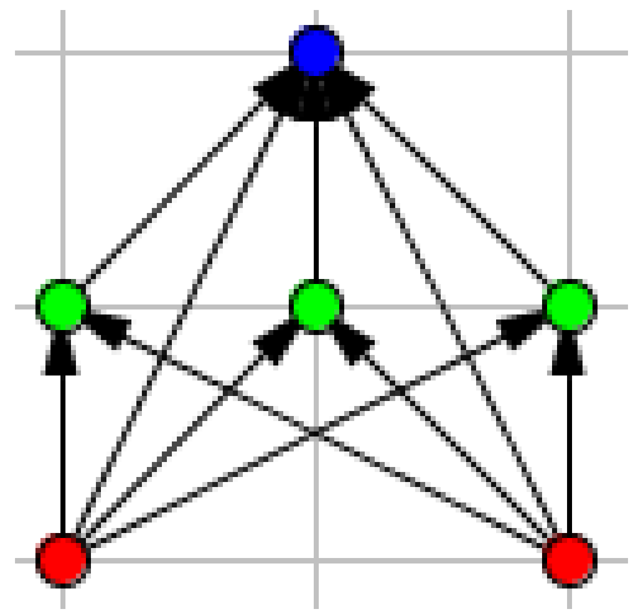
Abridged Problem Statement

- $N \leq 500$ rocks, each at (X_i, Y_i) and has slip rating of S_i
- Can climb from rock i to rock j if $Y_i < Y_j$ and $\max(|X_i - X_j|, |Y_i - Y_j|) \leq \max(S_i, S_j)$
- Find the smallest K such that for any set of K rocks, there is a directed path from some rock in the set to another
- K is hard to find so find K' instead
- K' = size of largest set of vertices such that no two vertices in the set are connected by a directed path
- It is easy to see that $K = K' + 1$
- In the right example, $K' = 2$
- Thus $K = 3$



Subtask 1

- $S_i = 2 \times 10^6$ for all i
- There is an edge from i to j as long as $Y_i < Y_j$
- Can only pick vertices with the same Y
- Count number of vertices with each value of Y using an array
- $K' =$ maximum number of vertices with the same Y
- For the right example, $K' = 3$
- Thus, $K = 4$



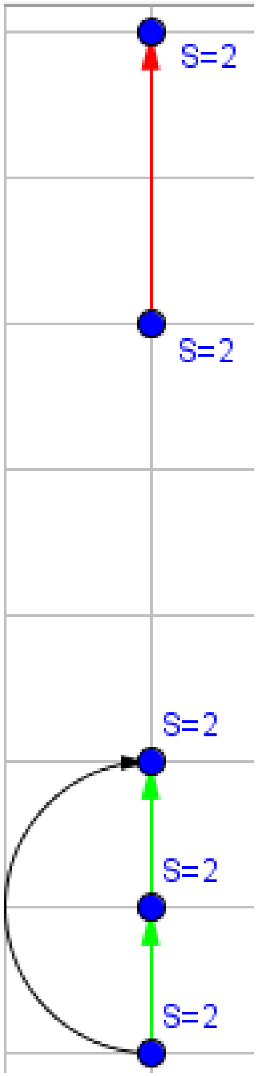
Subtask 2

- $1 \leq N \leq 20$
- N is small so check all $2^N \leq 2^{20} = 1048576$ possible subset of vertices
- For each subset, check every pair of vertices to see if there is a path from one to the other
- Use any search algorithm like DFS/BFS to search for path
- To make it faster, precompute for each pair of vertices whether there is a path from one to another and store in an array

Subtask 3

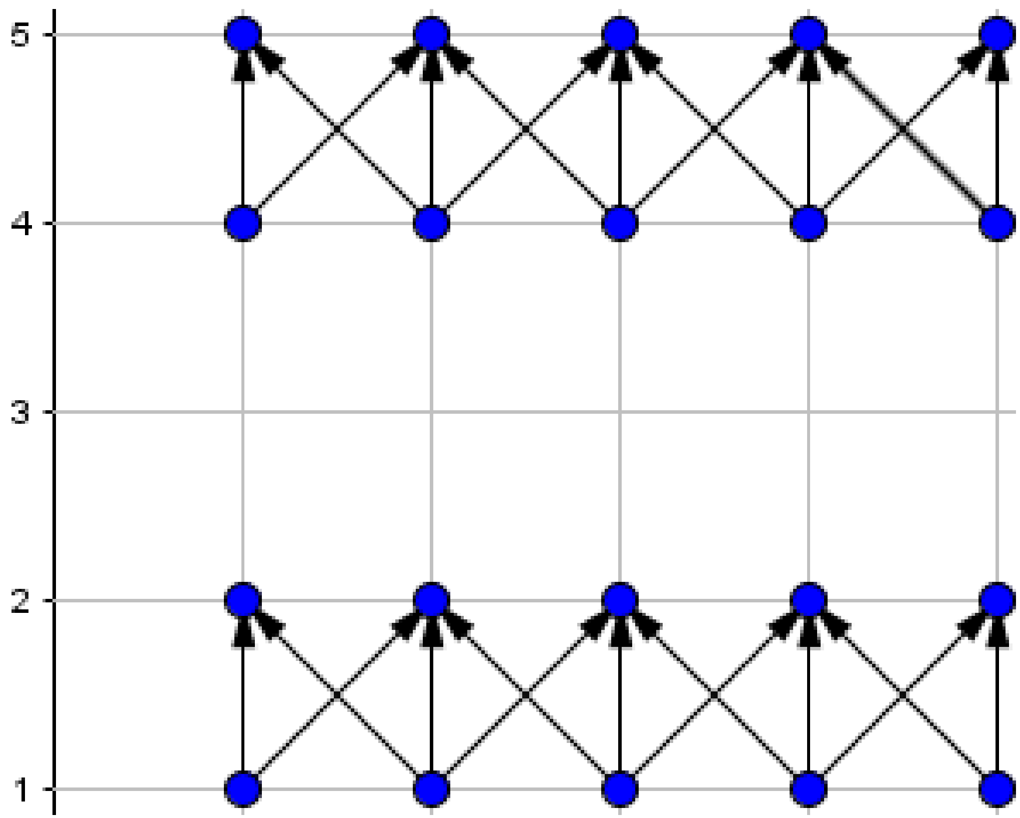
- $X_i = 0, S_i = S_j$ for all i and j
- For each vertex, check if there is an edge to the next highest vertex
- These edges will form disjoint paths, pick one vertex from each path
- $K' =$ number of such paths

  For the case on the right, $K' = 2$



Subtask 4

- $S_i = 1, Y_i$ is not a multiple of 3 for all i
- Note that this graph is bipartite

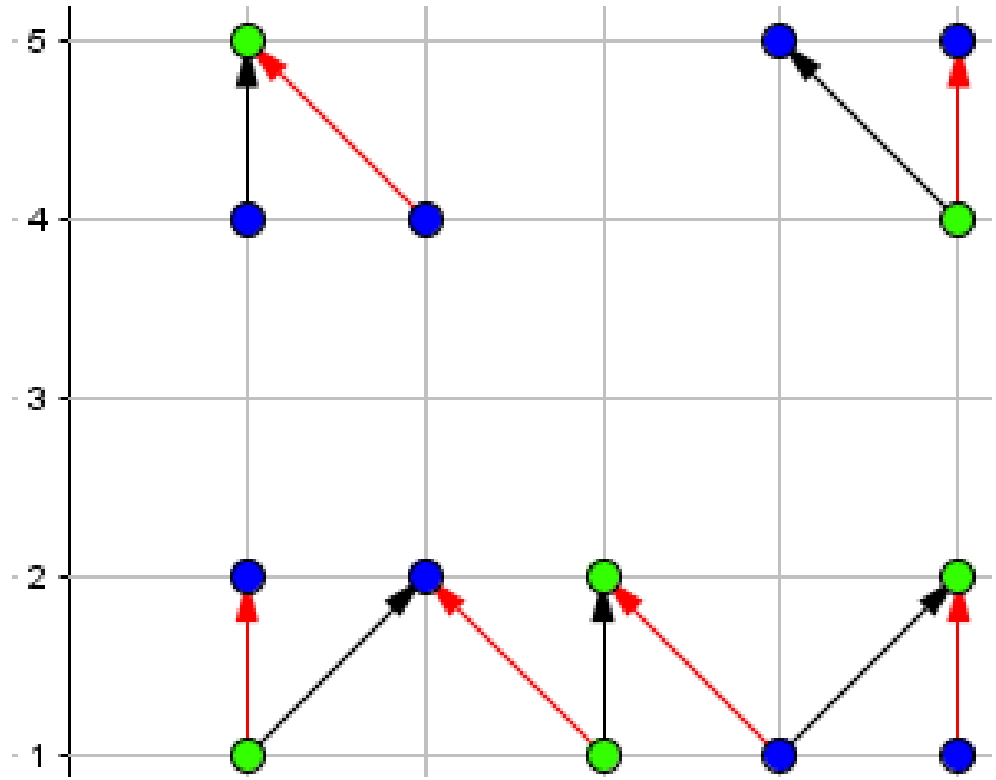


Algorithm

- Since all paths are of length 1, the answer is the maximum size of a set of vertices such that no 2 are connected by an edge
- => Maximum Independent Set (MIS) problem
- Size of MIS = $N -$ Size of Minimum Vertex Cover (MVC)
- As the graph is bipartite, Size of MVC $=$ Size of Maximum Cardinality Bipartite Matching (MCBM)
- Use $O(N^3)$ Augmenting Path Algorithm to find maximum matching of size M
- Answer is $N - M$

Example

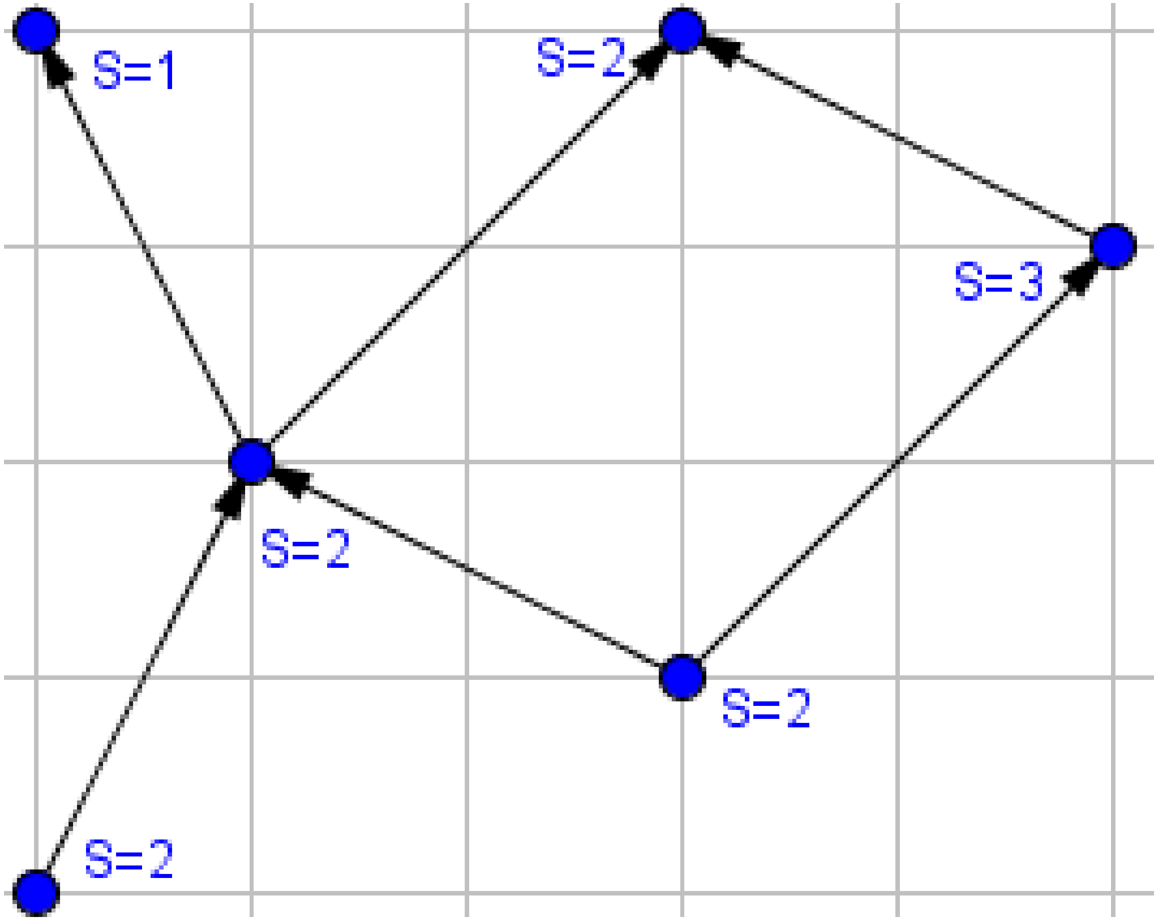
- Red edges form MCBM
- Green vertices form MVC
- Blue vertices form MIS
- For this case $K' = 6$
- Thus, $K = 7$



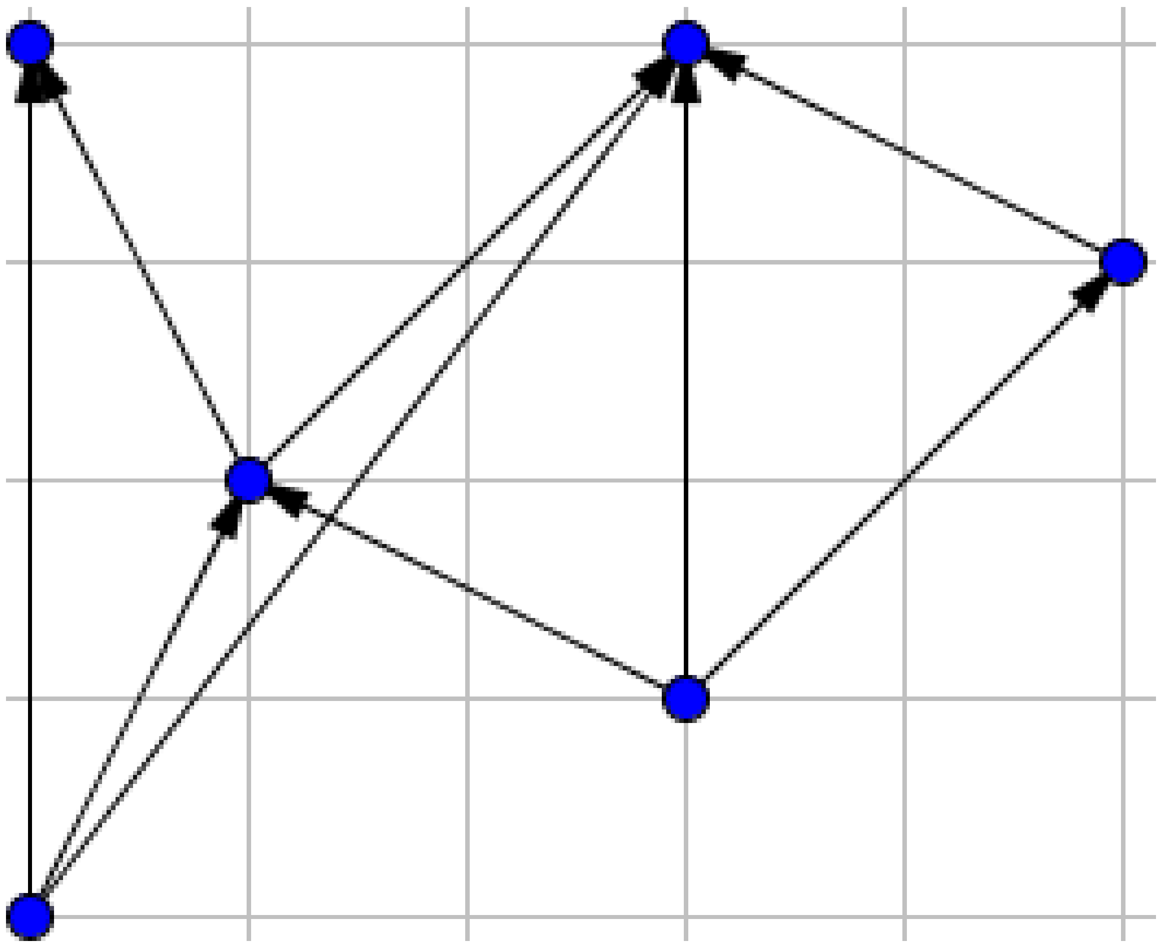
Subtask 5

- No other restrictions
- Construct a new graph where there is an edge from i to j if there is a path from i to j in the original graph
- To do this, set all the edge weights to 0 and run Floyd-Warshall algorithm
- Add an edge in the new graph from i to j if the shortest distance from i to j is 0
- Alternatively, can also run BFS/DFS from every vertex
- This is known as transitive closure

Example



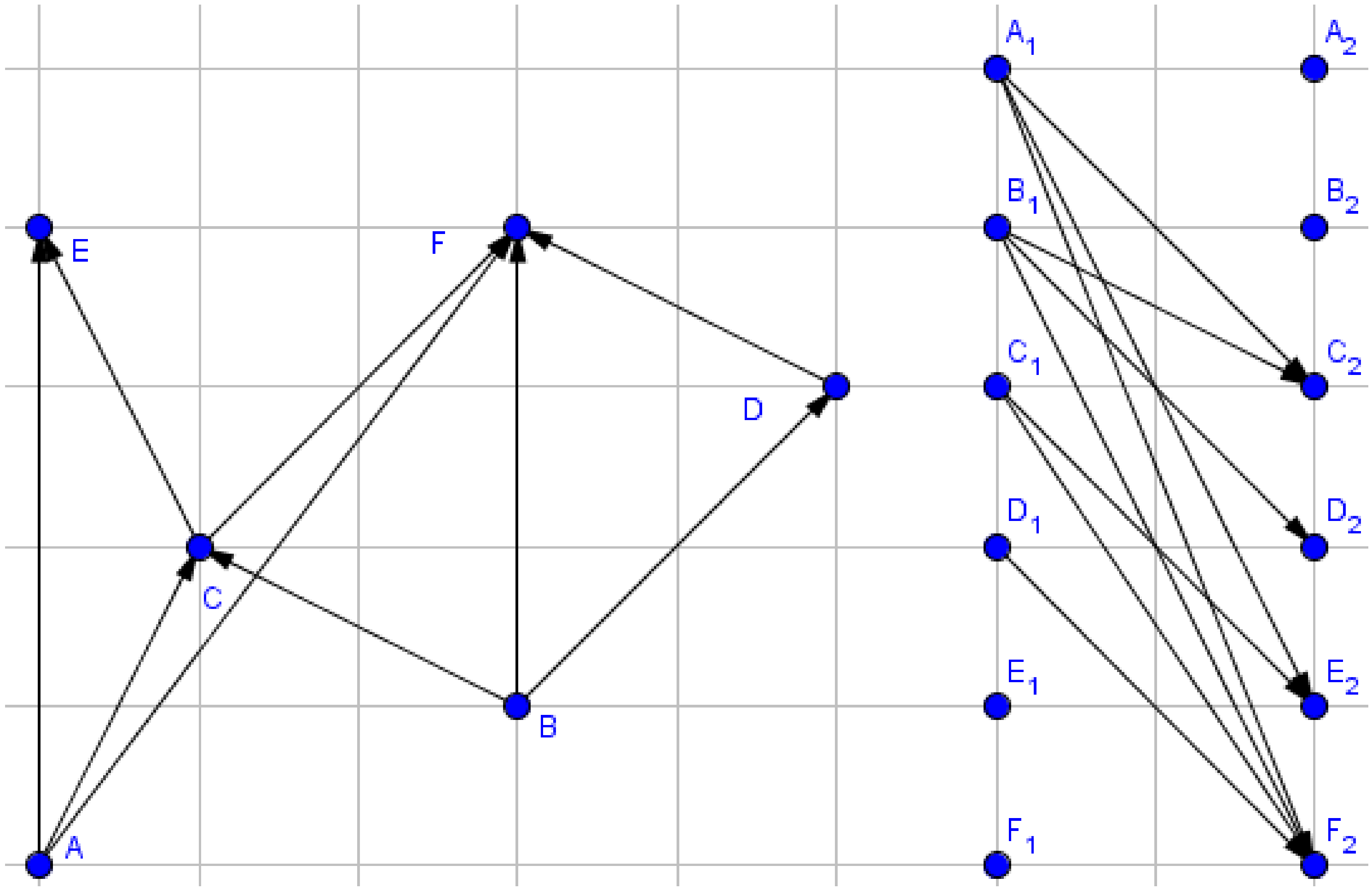
Transitive Closure



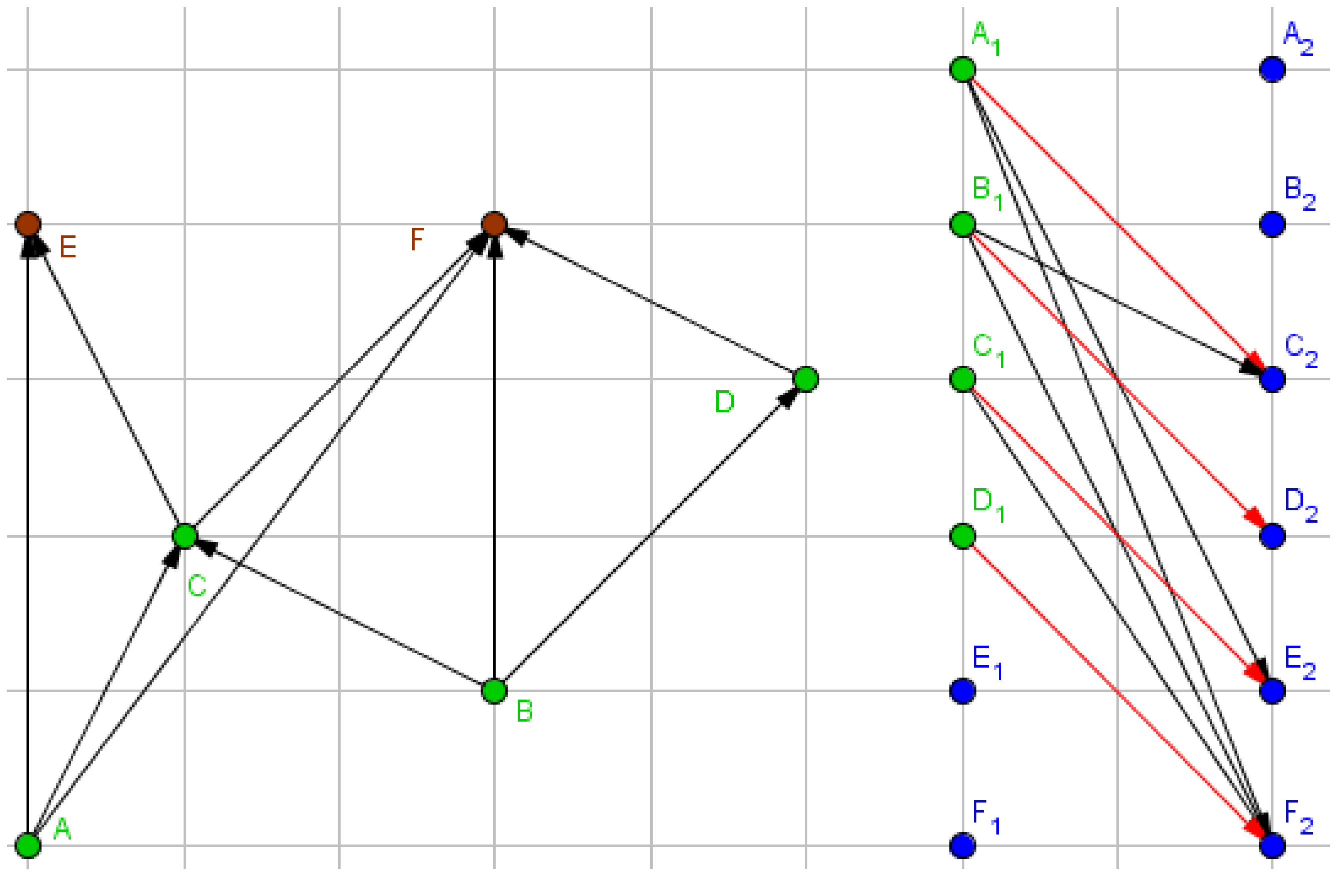
Algorithm

- In the new graph, an edge from i to j means there is a path from i to j so can't pick both
- => Maximum Independent Set (MIS) problem
- Construct a new graph with 2 copies of each vertex, i and i', add an edge from i to j' in the new graph for each edge from i to j in the original graph
- This graph is bipartite so Size of MVC = Size of MCBM
- It can be proven that the MVC in this bipartite graph will not contain both i and i'
- Thus, it equivalent to an MVC in the original graph
- Thus Size of MIS = $N -$ Size of MVC = MCBM

Bipartite Graph



Bipartite Matching



$$K' = 2 \Rightarrow K = 3$$

Conclusion

1. Perform transitive closure using Floyd-Warshall Algorithm
2. Construct bipartite graph with 2 copies of each vertex
3. Perform MCBM using augmenting path algorithm to get matching of size M
4. If $M = 0$, answer is -1, else $K = N - M + 1$

- Details of terms like MIS, MVC and MCBM can be found in Steven Halim's book "Competitive Programming 3"
- Proof of correctness of this algorithm is left as an exercise