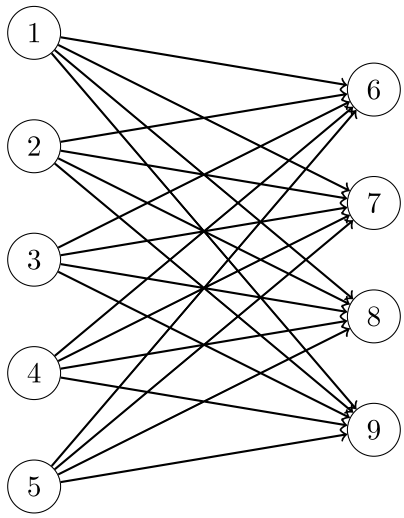
Problem 1

a.
The following is an outline of the graph for the problem:



However, the problem is infeasible as illustrated by the following:
Node 6 has a demand of 1530 units. Nodes 1,2,3,4 combined can only supply 805 units, while the arc from node 5 (which can supply 4031 units) to node 6, i.e. arc $(5, 6)$ has a capacity of 83 units. So, even if nodes 1,2,3,4 sent their whole supply to node 6 (i.e. 805 units), due to the capacity constraint, node 6's demand cannot be satisfied. The infeasibility can be verified using Gurobi

b.
We want to add capacity Δ_{ij} to arcs (i, j) to get a feasible flow. As we have a minimization problem, we want to add just enough extra capacity to get a feasible solution x. This gives us the following LP:

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in E} c_{ij}\Delta_{ij} \\
\text{subject to} \quad & \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = b_i \quad \forall i \in N \\
& x_{ij} - \Delta_{ij} \leq u_{ij} \quad \forall (i,j) \in E \\
& x \geq 0
\end{aligned}$$

We can solve for (Δ^*, x^*) using Gurobi. For the optimal solution we get, $\Delta^*_{28} = 144$, $\Delta^*_{56} = 1372$, $\Delta^*_{58} = 855$, while for x^* we get, $x^*_{18} = 208$, $x^*_{28} = 173$, $x^*_{29} = 20$, $x^*_{38} = 195$, $x^*_{46} = 75$, $x^*_{48} = 58$, $x^*_{49} = 76$, $x^*_{56} = 1455$, $x^*_{57} = 1583$, $x^*_{58} = 928$, $x^*_{59} = 65$. The optimal cost is $\sum c_{ij}\Delta_{ij} = 1,749,022$.

c.
Adding the computed Δ^* from part b to the capacities in the problem solved for part a we would get the following LP to solve for a min-cost flow:

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,j)\in E} c_{ij}x_{ij} \\ \text{subject to} \quad & \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = b_i \quad \forall i \in N \\ & x_{ij} \le u_{ij} + \Delta^*_{ij} \quad \forall (i,j) \in E \\ & x \ge 0 \end{aligned}$$

As one would expect, the optimal x^* comes out to be the same as the x^* of part b, because that is the only feasible flow. The optimal value is $\sum c_{ij}x_{ij} = 4,600,787$.

d.
Each flow in the graph only crosses 1 edge, thus each variable in x^* represents a flow. The flow decomposition would be the following:

Path	Flow
1-8	208
2-8	173
2-9	20
3-8	195
4-6	75
4-8	58
4-9	76
5-6	1455
5-7	1583
5-8	928
5-9	65

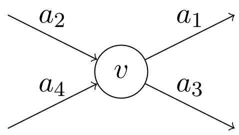
Problem 2

We show how to do this for a single node. The same procedure can then be repeated for every node. Take any node v that has a flow restriction of 10 units. In the original graph, replace v with 2 nodes, v' and v'' as follows:

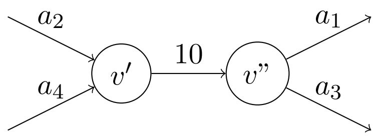
1. Insert an edge (v', v'') in E with capacity 10.
2. Replace every edge of the form $(u, v) \in E$ with (u, v'), keeping capacity same
3. Replace every edge of the form $(v, u) \in E$ with (v'', u), keeping capacity same

To illustrate, we give the following example:

Original graph:



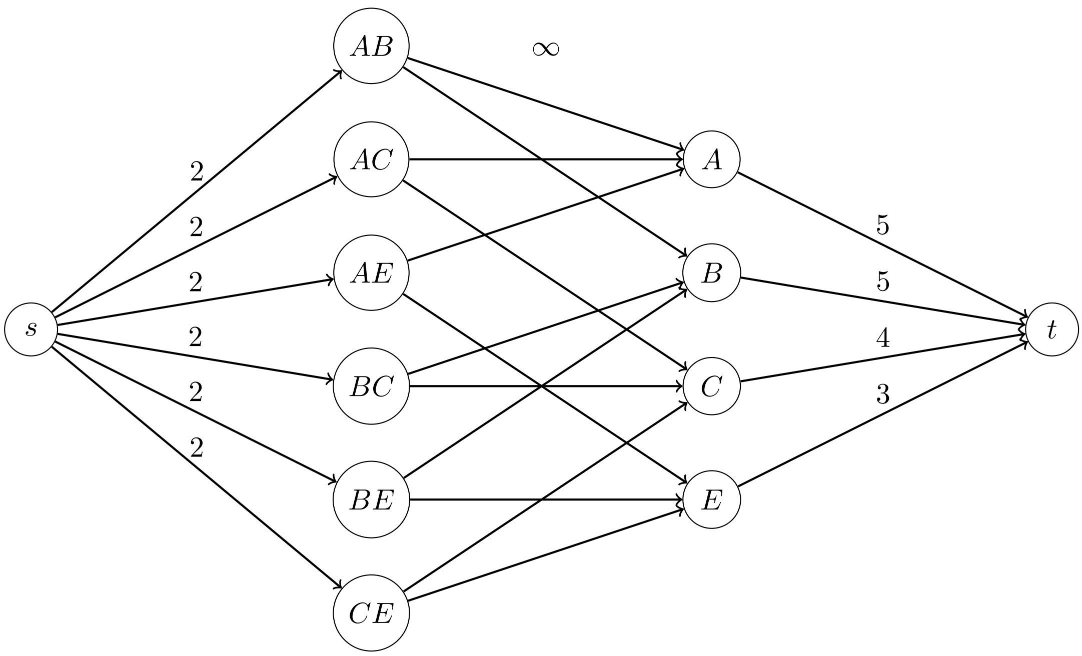
New Graph:



Problem 3

20 games remain to be played, where each pair of teams will play against each other twice. Each team has 8 games left to be played.

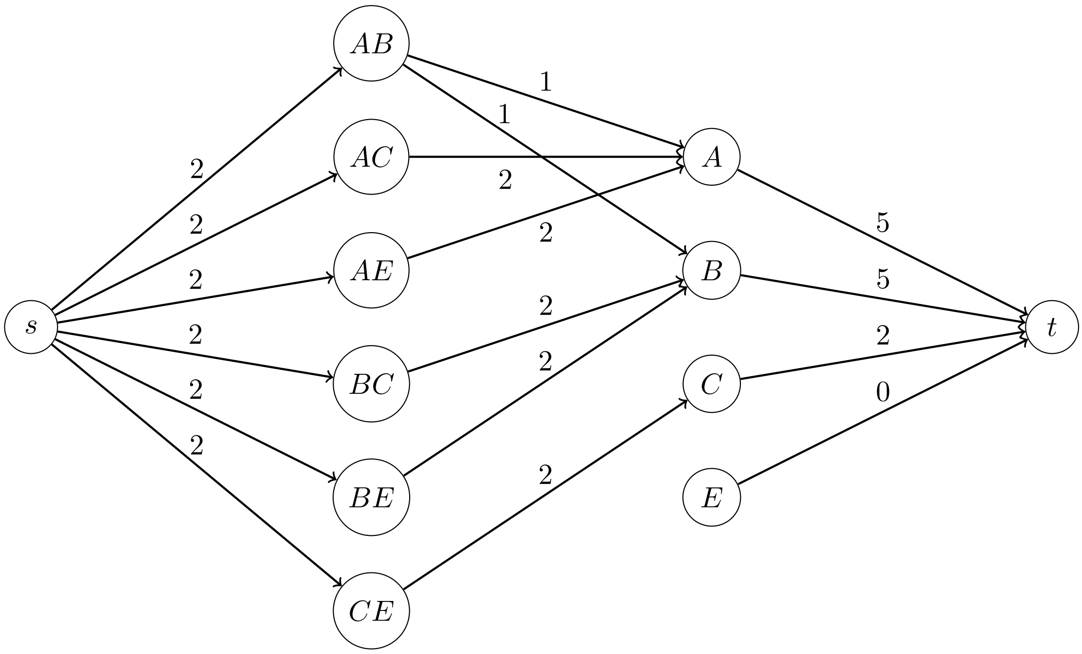
a. *D wins the tournament*:
First of all, we can clearly see that if D can win the tournament without winning all its remaining games, then it can certainly win the tournament while winning all games. So, we only try to see whether D can win the tournament *if it wins all games.* If D wins all its remaining games, it will win a total of 15 games. As long as no other team wins more than 15 games, D will have won the tournament. If we can find an assignment of results for the each of the remaining games such that no other team wins more than 15 games, then it means D can win the tournament. This problem can be expressed using the following graph:



In the above graph, the nodes on the left of the form IJ denote two teams that play together, and the ingoing arc denotes the number of games still to be played between teams I and J. From the 20 games that are left, we removed the 8 D is playing (since we are assuming it wins all of them), hence there are 12 games left. The nodes on the right denote a team, with the exclusion of D. A flow in this graph signifies a *flow of games*. Suppose there is a flow of 2 on the path $s \to AB \to A \to t$. This means that both the games played between A and B were won by A. A capacity for the edges of form (IJ, I) is not required, so we set all of them to ∞. Lastly, we want to limit the number of games won by each team such that they do not win more than 15, so the maximum number of games A is allowed to win is 5, for B also it is 5, for C it is 4 and for E it is 3. This is the motivation behind the capacities of the arcs going from a team to destination node t.

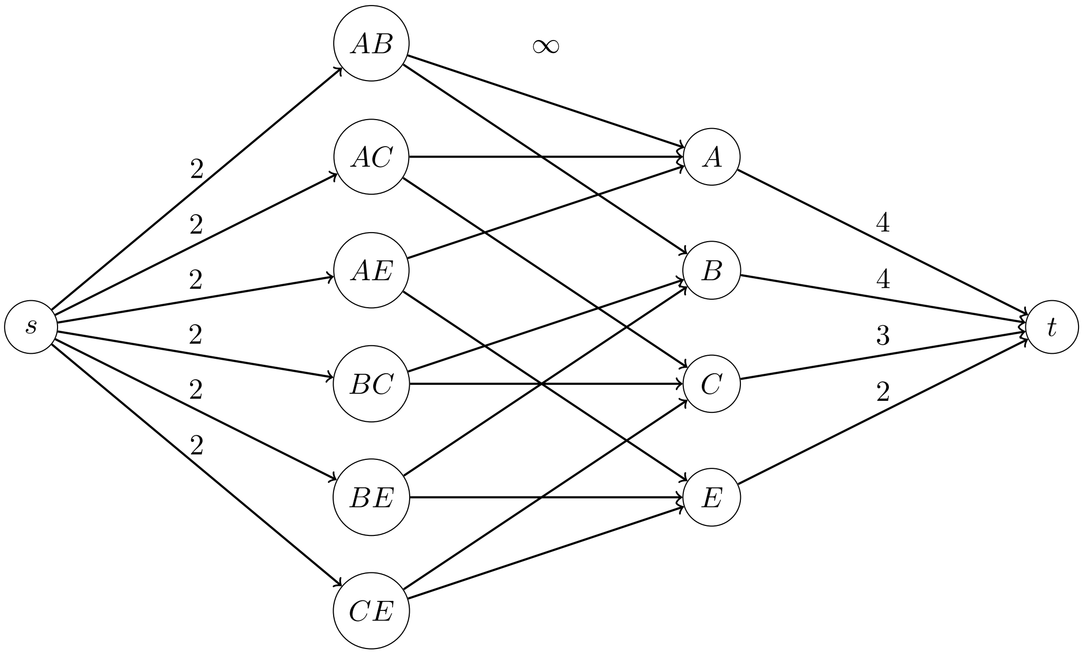
If there exists a flow of value 12 in the above graph, this means that there is a way of assigning the remaining games such that no team wins more than 15 games. Hence, a "YES" answer can be proved by exhibiting such a flow. On the other hand, if each feasible flow has value at most 11, there is no way D can win the tournament. By the max flow–min cut theorem, this can be certified by exhibiting a cut of value 11 or less. Hence, for a "NO" answer, it suffices to show such a cut. Note that solving the maximum flow problem over the graph with FF algorithm will allow us to find both a maximum flow and a minimum cut, hence the correct answer.

For the problem under consideration, a flow of 12 games indeed exists and is e.g. the following:

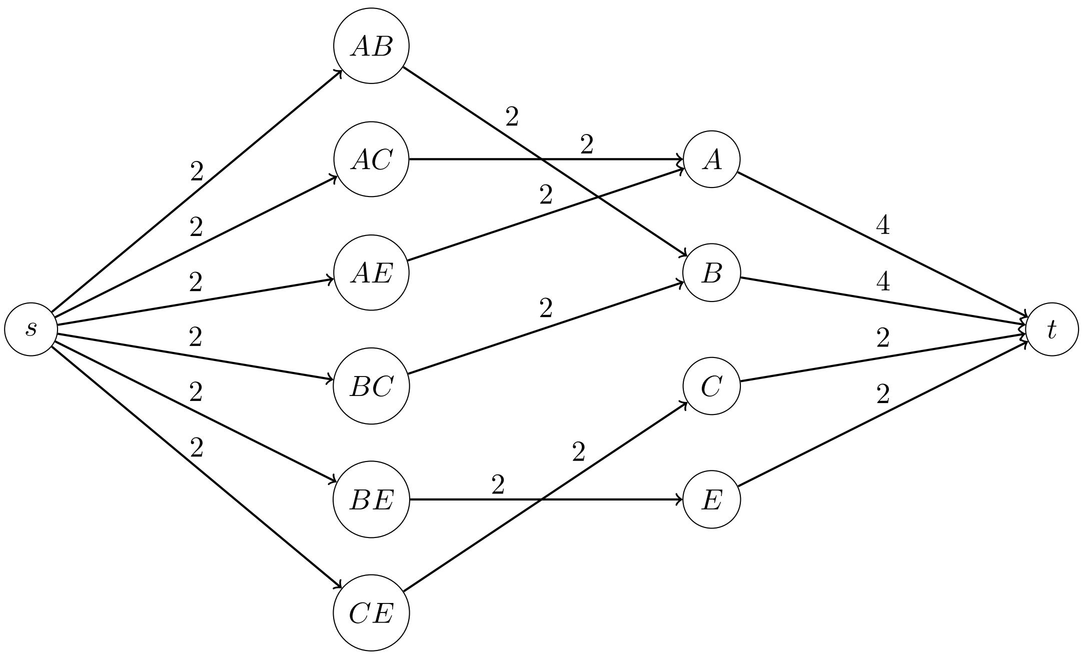


Therefore, D can still win the tournament.

b. *D is the only team that wins the tournament*:
The problem is essentially the same. We assume D wins all the games, so D wins 15 games. For D to be the only winner, each other team must win not more than 14 games. Thus, this time, the capacities of the last set of arcs will reduce by 1 and we will get the following graph:



Once again, we want to see if we can find a valid flow of 12 games in the graph or find a cut of value less than 12 to conclude that the flow is not possible. And once again, we can find a feasible flow:



Therefore, it is possible for D to be the only team to win.

Problem 4

1. As the robot can reach a particular cell from the cell above or the cell to the left, the total number of ways for a robot to reach (i, j) becomes:

$$f_i(j) = f_{i-1}(j) + f_i(j-1)$$

With initial value $f_1(1) = 1$. This gives us the following table of values for $f_i(j)$:

1	1	1	1	1	1	1
1	2	3	4	5	6	7
1	3	6	10	15	21	28
1	4	10	20	35	56	84

The value $f_4(7) = 84$ gives us the total number of ways for the robot to reach the bottom right cell.

2. Let $f_i(j)$ denote the maximum number of coins that can be collected if the board is of size $i \times j$. Then we get the following recurrence:

$$f_i(j) = \begin{cases} \max\{f_{i-1}(j), f_i(j-1)\} & \text{if no coin at } (i,j) \\ 1 + \max\{f_{i-1}(j), f_i(j-1)\} & \text{if coin at } (i,j) \end{cases}$$

with the initial value $f_1(1) = 0$.

0	0	0	0	0	1	1
0	0	1	2	2	2	3
0	1	1	2	3	3	3
0	1	2	2	3	4	4

Therefore, the maximum number of coins that can be collected is 4. To calculate the optimal path, we start from the bottom right corner, and go up or left in direction where $f_i(j)$ does not change until we cannot (i.e. when $f_i - 1(j) < f_i(j)$ and $f_i(j-1) < f_i(j)$). At this point, we move in the direction (up or left) which decreases value by 1 (if value decreases by 1 in both directions we choose randomly). We continue doing this until we reach the top left corner, and that gives us the reverse of an optimal route. In the current question an optimal route is:

$$(1,1) \to (1,2) \to (2,2) \to (3,2) \to (4,2) \to (5,2) \to (5,3) \to (6,3) \to (6,4) \to (7,4)$$