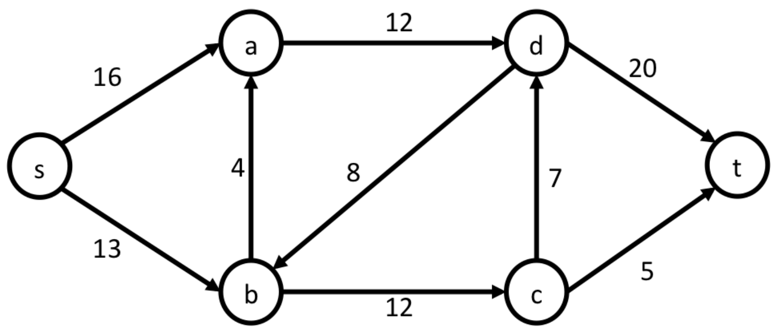
- This homework is optional and will not be graded.

There are 7 questions for a total of 0 points.

---

1. Consider the s-t network graph G below:



Let f be the following mapping from edges to numbers:

f(s, a) = 12	f(s, b) = 5	f(a, d) = 12	f(b, a) = 0	f(b, c) = 12
f(d, b) = 7	f(d, t) = 12	f(c, d) = 7	f(c, t) = 5	

Answer the following questions with respect to the network graph G and mapping f.

(a) State true or false: f is an s-t flow.

(a) ______________

(b) State true or false: f is an s-t flow with maximum value.

(b) ______________

(c) Give an s-t cut with minimum capacity of the graph above.

(c) ______________

2. Answer the following :

(a) State true or false: A perfect matching exists in any bipartite graph $G = (L, R, E)$ such that $|L| = |R|$ and all vertices in G have degree ≥ 2.

(a) ______________

(b) Give a reason for your answer in part (a).

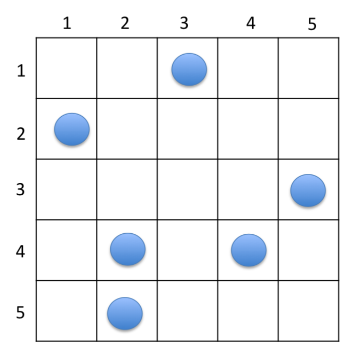
3. Answer the following :

(a) State true or false: A perfect matching exists in any bipartite graph $G = (L, R, E)$ such that $|L| = |R|$ and all vertices in G have degree exactly equal to d, an integer ≥ 2.

(a) ______________

(b) Give a reason for your answer in part (a).

4. (*Placing balls on board*) You are given an $n \times n$ board. In a few cells of the board, there is a ball present. The location of a ball may be given using the $(row, column)$ pair.
For example, consider the following 5×5 board given in the left side of the figure below. There are balls present in locations $(1,3),(2,1),(3,5),(4,2),(4,4),(5,2)$. A way to represent this information is a 5×5 0-1 matrix which has 1 in the cell (i,j) iff there is a ball in the location (i,j) on the board. The figure on the right gives the 0-1 matrix representing locations at which the balls are on the board.



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

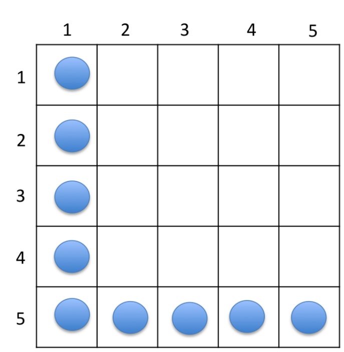
Figure 1: Example board (left) and 0-1 matrix representing information (right).

Two balls on the board are said to *intersect* if and only if they share the same row or same column. (*For example, balls at locations $(4,2)$ and $(5,2)$ in the figure intersect since they share the same column.*) Your goal is to design an algorithm to determine if picking a subset S of n balls from a given board is possible so that no two balls in S intersect. Answer the questions below:

(a) Give a subset of 5 balls from board given in Figure 1 such that no two balls in the subset intersect. You may write this subset as a list of locations of the five balls.

(a) ____________________

(b) Consider the 5×5 board below.



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

Figure 2: Another example

State true or false: It is possible to pick a subset of 5 balls from the board given in Figure 2 such that no two balls in the subset intersect.

(b) ____________

(c) Give pseudocode for an algorithm that takes as input n and a 0-1 matrix M representing the location of balls on an $n \times n$ board and outputs "yes" if it is possible to pick a subset S of n balls from a given board such that no two balls in S intersect. Otherwise, it outputs "no". Discuss the running time of your algorithm.

5. (*Hiring problem*) n companies come for campus recruitment to IITD, with each company looking to hire exactly two graduates. There are t graduates this year participating in the campus recruitment. The eligibility information of graduates for jobs in companies is given in the form of a $n \times t$ matrix M. The entry (i, j) of the matrix (that is $M[i, j]$) is 1 if the j^{th} candidate is eligible for getting hired by the i^{th} company and it is 0 otherwise.
(*Consider the example below where there are* 7 *candidates and* 3 *companies. In this scenario, Candidate* 1 *is eligible for getting hired in company* 1 *but is not eligible for getting hired by companies* 2 *and* 3.)

	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	0	1	0	0	0	0	1
3	0	0	1	0	1	1	0

Figure 3: An Example recruiting scenario.

Your goal in this problem is to design an algorithm to determine if all companies can hire exactly two graduates each such that no graduate is simultaneously hired in more than one company.

(a) In the above example scenario in Figure 3, give the graduates hired by the three companies such that all companies hire exactly two graduates and no graduate is simultaneously hired by more than one company.

Company	1	2	3
Graduates			

(b) Consider the scenario below with 6 graduates and 3 companies.

	1	2	3	4	5	6
1	1	0	1	0	1	0
2	1	0	1	1	0	0
3	0	1	0	0	1	1

<u>State true or false</u>: In the above scenario, all three companies can hire exactly two graduates such that no graduate is simultaneously hired by more than one company.

(b) ______________

(c) Give pseudocode for an algorithm that takes as input n, t and a 0-1 matrix M representing the eligibility information and outputs "yes" if all companies can hire exactly two graduates such that no graduate is hired by more than one company. Otherwise, it outputs "no". Discuss the running time of your algorithm.

6. (*MIS of bipartite graph*) Design an algorithm for finding a maximum independent set of a given bipartite graph $G = (L, R, E)$.

7. (*Another algorithm for max-flow*) Consider the following slightly changed version of the Ford-Fulkerson max-flow algorithm. This algorithm is also due to Jack Edmonds and Richard Karp.

Max-Flow

- Start with a flow f such that $\forall e \in E, f(e) = 0$.
- While there is an $s - t$ path in G_f
    - Find an $s - t$ path P in G_f with **largest bottleneck value**.
        - Augment along P to obtain f'.
        - Update f to f' and G_f to $G_{f'}$

return(f).

---

1. Think of an algorithm to find the *largest bottleneck path* from s to t in a given graph. A bottleneck path is a path such that the bottleneck edge has maximum weight. Discuss its running time. (*Hint: Try ideas from Dijkstra's Algorithm.*)
2. Let f be any $s - t$ flow and t be the value of maximum flow in the residual graph G_f. Let f' be the new flow after one augmentation and t' be the value of the new maximum flow in the residual graph $G_{f'}$. Argue that $t' \leq (1 - 1/m) \cdot t$.
3. Use the properties you showed above to argue that for a graph with integer capacities, the algorithm runs in time $O(m^2 \cdot \log m \cdot \log f^*)$, where f^* is the value of the max-flow in the original graph G.