Quiz 2

- Do not open this quiz booklet until you are directed to do so. Read all the instructions first.
- The quiz contains 6 problems, several with multiple parts. You have 2:15 hours for this midterm.
- This booklet contains a total of 16 pages, including this one and three sheets of scratch paper.
- Write your solutions in the space provided. If you run out of space, continue on a scratch page and make a notation.
- When we ask you to give an algorithm in this quiz, describe your algorithm in English or pseudocode, and provide a short argument for correctness and running time. You do not need to provide a diagram or example unless it helps make your explanation clearer.
- Do not waste time deriving facts that we have studied. Just cite results from class.
- Do not spend too much time on any one problem.

♣ **Do not remove any of these pages!**

♣ **Please write your name on every single page of this exam.**

Problem	Title	Points	Parts	Grade	Initials
0	Name	1	1		
1	True/False	8	4		
2	Short Answers	12	2		
3	Maximum Vertex Biclique	20	2		
4	Correcting Flows	20	2		
5	Linear Program for Matchings	20	3		
6	Sum of Minima	20	1		
Total		100			

NAME: _____

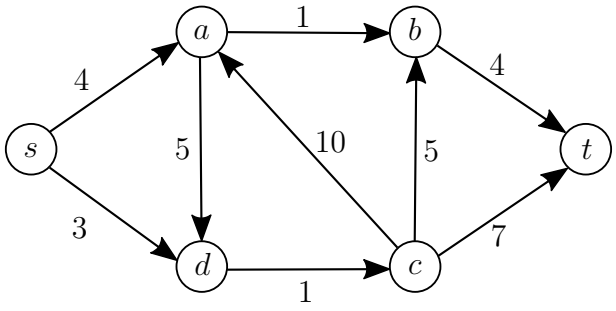
Circle your recitation:

F10	F11	F12	F1	F2	F3	F11	F12	F1
Katerina	Themis	Shankha	Maryam	Nishanth	Prashant	Manolis	Shalom	Ethan
R01	R02	R03	R04	R05	R06	R07	R08	R09

Quiz 2-1: [8 points] True/False Questions: Parts (a) – (d)

Mark each statement as either true or false. You have to **provide a very short explanation for each**.

(a) [2 points] In the following network, it is possible to send 3 units of flow from node s to node t.



(b) [2 points] A problem $\Pi \in NP$ if and only if there exists a polynomial time algorithm V_Π such that: *For all* x, $\Pi(x) = yes$ *if and only if there exists a string* y *such that* $V_\Pi(x, y) = 1$.

(c) [2 points] If Π_1 can be reduced to Π_2 and Π_2 is NP-complete, then Π_1 is NP-complete.

(d) [2 points] Consider the 2-player game where each player has 3 possible strategies and has the following payoff matrix (the bottom left corner denotes the payoff of the Player B and the upper right corner denotes the payoff of Player A) :

Player B \ Player A	1	2	3
a	0 0	+1 -1	-2 +2
b	-1 +1	0 0	+1 -1
c	+2 -2	-1 +1	0 0

The mixed strategy where

i. A chooses one of the strategies 1, 2, 3 with probability 1/3 each.

ii. B chooses one of the strategies a, b, c with probability 1/3 each.

is a Nash equilibrium of the described game.

Quiz 2-2: [12 points] Short Answers: Parts (a) – (b)

For each of the following questions you have to **provide an answer together with a short proof for your statement**.

(a) [6 points] A *planar graph* is a graph that can be drawn on the plane in such a way that its edges intersect only at their endpoints (that is, when drawn as straight line segments the edges do not cross each other). Given an undirected weighted planar graph $G = (V, E)$, can you solve All-Pairs-Shortest-Paths in G in $O(|V|^2 \log |V|)$ time?
Hint: You can use without proof that any planar graph has at least one vertex with degree at most 6.

(b) [6 points] Assume f is convex, and consider the following condition on $\vec{x}$, where $\vec{x} \in \mathbb{R}^n$:

$$\forall \vec{d} \in \mathbb{R}^n, \quad \textit{it holds that} \quad \langle \nabla f(\vec{x}), \vec{d} \rangle \geq 0 \quad (*)$$

We also remind you the convexity condition in high dimensions, i.e. $n > 1$. A function f is *convex* if and only if

$$\forall \vec{x}, \vec{y} \in \mathbb{R}^n, \quad f(\vec{y}) \geq f(\vec{x}) + \langle \nabla f(\vec{x}), \vec{y} - \vec{x} \rangle$$

Where we use $\langle \cdot, \cdot \rangle$ to denote the *dot product* in $\mathbb{R}^n$.
Explain with words why $(*)$ holds if and only if $\vec{x}$ is optimal, i.e. this is an *optimality condition*.

Quiz 2-3: [20 points] Maximum Vertex Biclique: Parts (a) – (b)

We define the *biclique* $K_{a,b}$ with parameters a, b to be a bipartite graph with a vertices on one side, b vertices on the other side and edges between every vertex on one side to every vertex on the other side. More precisely

$$K_{a,b} = (A \cup B, \{\{u, v\} \mid u \in A, v \in B\}) \text{ with } |A| = a, \ |B| = b$$

We remind you that an *induced subgraph* G_S of a graph G is a graph that contains a subset S of the vertices of $G = (V, E)$ together with all edges whose endpoints are both in S. Formally

$$G_S = (S, \{\{u, v\} \mid \{u, v\} \in E \ \textit{and} \ u, v \in S\})$$

We also define an *empty graph* I_c with parameter c to be the graph

$$I_c = (S, \varnothing) \ \textit{with} \ |S| = c$$

An *independent set* of a graph $G = (V, E)$ is an induced subgraph of G that is an empty graph, i.e. an induced subgraph of G with c vertices and no edge between them.

Given an unweighted, undirected bipartite graph $G = (V = V_1 \cup V_2, E)$, our goal is to find the size of the induced subgraph of G that is a biclique with the maximum number of vertices possible. For convenience we let $n = |V|$ and $m = |E|$.

(a) [10 points] Let p be the size of the minimum vertex cover in G. Prove that the size of the maximum independent set in G is $n - p$.

(b) [10 points] Provide an algorithm that given G as input finds the size (number of vertices) of the induced subgraph of G with the maximum number of vertices that is a biclique. The running time of your algorithm should be $O(n^{5/2})$.
Hint: If $H = (V_H, E_H)$ is a simple undirected graph, then we define the *complement* of H, denoted by $\overline{H}$, to be the graph

$$\overline{H} = (V_H, \overline{E}_H = \{\{u, v\} \mid u, v \in V_H \text{ and } \{u, v\} \notin E_H\})$$

Define the bipartite version of the complement operation and use it along with part (a).

Quiz 2-4: [20 points] Correcting Flows: Parts (a) – (b)

Let $G = (V, E, c)$ be a directed graph with *integer* capacities $c : E \to \mathbb{N}$. We also fix a source $s \in V$ and a sink $t \in V$. Suppose that you are given a valid flow for graph G , $f_G : E \to \mathbb{N}$, such that f_G pushes the maximum possible amount of flow from s to t. That is, f_G is a max flow in G.

Suddenly something happens to the flow network and the capacity $c(e_1)$ of one specific edge $e_1 \in E$ changes. We define the new capacity function $c' : E \to \mathbb{N}$ to be identical to c for every edge except for edge e_1, which changes to the new capacity $c'(e_1) \neq c(e_1)$. We also define $H = (V, E, c')$. Now we are no longer sure whether f_G is a valid flow, and even if it is valid we are not sure that it is the optimal flow. Our goal is to find the new optimal flow f_H as quickly as possible, given G, H and f_G. For simplicity we set $n = |V|$, $m = |E|$ and $L = |c'(e_1) - c(e_1)|$. We denote by G_f the residual graph of G after sending flow f.

(a) [8 points] If we have the guarantee that $f_G(e_1) \leq c'(e_1)$, provide an algorithm that finds f_H given G, H, f_G. Prove that the running time of your algorithm is $O((n + m)L)$.

Now we assume that $c(e_1) \geq |f_G(e_1)| > c'(e_1)$.

(b) [12 points] Given f_G provide an algorithm that finds f_H (i.e the maximum flow in the new graph) in time $O((n+m)L)$.
Hint: As a first step find an algorithm that changes f_G to a valid flow for H. To do so find an algorithm that is given the residual graph G_{f_G} and pushes back at most L units of flow from t to s in a way that all this flow goes through the e_1 in the reverse direction.

Quiz 2-5: [20 points] Linear Program for Matchings: Parts (a) – (c)

Given a simple undirected graph $G = (V, E)$ our goal is to see how we can use linear programs to compute maximum matchings in G. Note that G need not be bipartite. For any vertex $v \in V$ we define the neighborhood of v

$$\Gamma(v) = \{u \in V \mid \{u, v\} \in E\}$$

(a) [5 points] Using one variable x_e for every edge e, we formulate the maximum matching problem as the following integer linear program (ILP1)

$$\begin{aligned} & \max \sum_{e \in E} x_e \\ \text{(ILP1)}: \qquad & \textit{s.t.} \sum_{u \in \Gamma(v)} x_{\{u,v\}} = 1 \quad \forall v \in V \\ & x_e \in \{0, 1\} \end{aligned}$$

The formulation given above has a small mistake.

Correct the mistake in the above ILP (ILP1) to get another ILP (ILP2) that is the correct integer linear program formulation of the maximum matching problem. Also relax the integrality constraints of both (ILP1) and (ILP2) to get the corresponding linear programming relaxations (LP1) and (LP2) that might have fractional solutions.

For the rest of the problem, except for the last part, we will only consider *2-regular* graphs. This means that every vertex in G has degree exactly 2.

(b) [7 points] Given the optimal solution x^*_e of (LP1)[1] we define $E_1 = \{e \mid x^*_e \neq 1/2\}$ and $E_2 = \{e \mid x^*_e = 1/2\}$. Prove that both the graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ consist of disjoint cycles.

Based on this show how you can change x^* to an optimal solution $\hat{x}$ such that for any $e \in E$, $\hat{x}_e \in \{0, 1/2, 1\}$.

[1] Although our goal is to deal with (LP2) you can show that it is the same to work with (LP1) in the case of 2-regular graphs.

Now we turn our attention to general graphs that might not be 2-regular. For an optimal solution $\hat{x}$ we again define $E_1 = \{e \mid \hat{x}_e \neq 1/2\}$ and $E_2 = \{e \mid \hat{x}_e = 1/2\}$ and $G_1 = (V, E_1)$, $G_2 = (V, E_2)$.

(c) [8 points] We give you an unweighted graph $G = (V, E)$ and an optimal solution $\hat{x}$ of (LP2) such that for all $e \in E$, $\hat{x}_e \in \{0, 1/2, 1\}$. Also assume that G_2 contains **at most one odd cycle**. Provide an algorithm with running time $O(|V| + |E|)$ that given $\hat{x}$ finds a maximum matching in G.
Hint: As a first step prove there is an optimal matching that does not contain any edge $e \in E$ with $x_e = 0$ and contains all the edges $e \in E$ with $x_e = 1$.

Quiz 2-6: [20 points] Sum of Minima

We are given an array $A[1 \ldots n]$ such that $A[i] \geq 0$ for any i. Our goal is to compute the quantity

$$B = \sum_{i=1}^{n} \sum_{j=i}^{n} \min_{i \leq k \leq j} A[k]$$

For example, if $A = [3, 1, 2]$ then $B = 9$. Provide an algorithm to compute B. For this problem the credit depends on the running time of your algorithm. Here is the list of the running times together with their credit.

Advice: In case you are going for option **iv.**, we suggest you finish the rest of the exam before.

i. [2 points] For an algorithm with running time $O(n^3)$.

ii. [15 points] For an algorithm with running time $O(n^2)$.

iii. [18 points] For an algorithm with running time $O(n \log n)$.

iv. [20 points] For an algorithm with running time $O(n)$.

Scratch page