CS 0368-4246: Combinatorial Methods in Algorithms (Spring 2025) July 23, 2025

Final Exam (A)

Instructor: *Or Zamir*

Instructions (Please Read Carefully)

- This exam consists of **four** questions. The number of points for each question and sub-question is indicated alongside it. The total available points add up to **105**, but the final score will be capped at **100**.
- You may use theorems or lemmas that were proved in class, provided you cite them exactly as stated. Any modifications of these results—or use of results from homework—must be fully re-proven.
- Write your answers clearly and in an organized manner in the provided exam booklet. Clearly label each answer with the corresponding question number, and indicate whether each page is a draft or a final answer.
- Unless explicitly stated otherwise, all graphs are assumed to be undirected and unweighted, and all notation follows that used in class.
- Unless explicitly stated otherwise, algorithms with an expected run-time or that succeed with high probability are sufficient.

Exam Questions

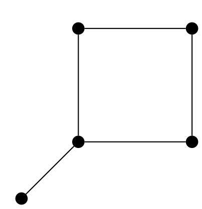

Figure 1: The kite graph $\mathcal{K}$.

1. *(40 points)* Denote by $\mathcal{K}$ a graph on five vertices we would call *a kite*, this is a 4-cycle C_4 with an additional node connected to exactly one of the cycle vertices, see Figure 1.

   (a) *(15 points)* Show that given a graph G with n vertices, we can check if it contains a kite $\mathcal{K}$ as a subgraph in $\tilde{O}(n^2)$ time.

   (b) *(10 points)* Assume that there is no algorithm for C_4-detection in n-vertex graphs running in $n^{2-\varepsilon}$ time for any constant $\varepsilon > 0$. Prove that no algorithm can detect a kite in n-vertex graphs in time $n^{2-\varepsilon}$ for any constant $\varepsilon > 0$.

   (c) *(15 points)* What is the extremal number of $\mathcal{K}$? Give tight, up to constant factors, upper and lower bounds for $ex(\mathcal{K}, n)$.

2. *(40 points)*

   (a) *(15 points)* Let G be a connected graph and φ a parameter. Assume that for any subset of edges $D \subseteq E(G)$ if we denote by $C_1, \ldots, C_r$ the connected components of $G - D$ for which $vol(C_i) \leq \frac{1}{2}vol(V(G))$, it holds that $\sum_{i=1}^{r} vol(C_i) \leq |D|/\varphi$. Prove that the conductance of G is at least $\Omega(\varphi)$.

   (b) *(15 points)* Let G be a graph without isolated vertices, prove that $\lambda_2(N_G) > 0$ if and only if G is connected. Note that we proved the same statement in class for the Laplacian L_G rather than for the normalized Laplacian $N_G := D_G^{-1/2} L_G D_G^{-1/2} = I - D_G^{-1/2} A_G D_G^{-1/2}$.

   (c) *(10 points)* Strengthen the previous statement and show that for every connected G with n vertices we have $\lambda_2(N_G) \geq \Omega(1/n^4)$. (Note: this is not necessarily a tight bound).

3. *(15 points)* Let G be a *weighted* graph. A **MinCut query** is an algorithm that, given a subset $S \subseteq V$, returns the minimum-weight edge with one endpoint in S and one in $V \setminus S$, if such an edge exists. You are given the vertex set of G, but have no other access to the graph (in particular, you cannot check whether an edge exists, or query degrees or weights). Your access is only through MinCut queries. Design an algorithm that computes a minimum spanning tree of G using as few MinCut queries as possible. Provide a full analysis, but there is no need to prove your solution is optimal.

4. *(10 points)* Let G be graph with n vertices. Give a $O^\star(2^n)$ time algorithm that finds the smallest number of cliques (of any sizes) needed to cover all vertices of G (this is a collection of cliques in G such that the union of their vertex sets is V).

Solution

Question 1.

a) We observe that every C_4 in which one of the vertices has degree at least 4 is contained in a $\mathcal{K}$ — that is because the high-degree vertex has at least one neighbor which is not part of the C_4 itself. A slight modification of the $O(n^2)$ algorithm we used to find a C_4 in class can also find a C_4 with a vertex of degree at least 4 if one exists in the graph:

- Go over all vertices and for each one create an additional neighbor list $N_{\geq 4}(v) \subseteq N(v)$ that contains all of v's neighbors for which the degree is at least 4.
- Initiate a dictionary or set D maintaining pairs of vertices.
- Iterate over all vertices v in the graph:
  - Go over all pairs u, w of v's neighbors such that $u \in N_{\geq 4}(v)$ and $w \in N(v)$ (make sure to go over each pair only once).
  - For each such pair, check if it is already in D. If so, return "Yes", otherwise, insert it to D.
- Return "No" unless we returned "Yes" beforehand.

As in class, this algorithm maintains a set of the endpoints of distinct 2-paths, if we have a collision we found a C_4, and we may add at most $n^2/2$ different endpoints before we find a collision. This time, we only consider endpoints such that at least one of them is of degree at least 4 - thus we would find a C_4 if and only if at least one of its endpoints is such a vertex.

Next, we observe that the only copies of $\mathcal{K}$ that the above algorithm may miss are those where *all* C_4 *vertices* are of degree at most 3. To check whether such a copy exists we remove all vertices of vertices of degree at least 4 from the graph and then naively write down *all* C_4s in the resulting graph which is of maximum degree 3. This can be done by going over the vertices and for each one writing down its distance 2 neighborhood (which is of size $O(1)$). Then, for each of these $O(n)$ C_4s we check whether or not they are in a $\mathcal{K}$ (that is, one of their vertices has a neighbor outside of the cycle itself).

b) We reduce the problem of finding a C_4 in a graph to the problem of finding a $\mathcal{K}$ in a graph. Given an input graph G we add an auxiliary vertex v' and connect it (only) to v for each vertex v. Call the resulting graph G'.

We observe that G' contains a $\mathcal{K}$ if and only if G contains a C_4: If G contains a C_4 and v is an arbitrary vertex within it, then the same C_4 in addition to v' in G' is a $\mathcal{K}$. On the other hand, if G' contains a $\mathcal{K}$ then it also contains a $C_4 \subseteq \mathcal{K}$ and every vertex within this four-cycle is of degree ≥ 2 and thus is not an auxiliary vertex — in particular, that C_4 also exists in G.

As the number of vertices in G' is $2n$, an algorithm finding a $\mathcal{K}$ in $n^{2-\varepsilon}$ time results in an algorithm to find a C_4 in $(2n)^{2-\varepsilon}$ time. This is a contradiction to our assumption.

c) As stated before, every $\mathcal{K}$ contains a C_4 and thus a C_4-free graph is also $\mathcal{K}$-free. This implies that $ex(\mathcal{K}, n) \geq ex(C_4, n) = \Omega(n^{3/2})$ where we proved the last inequality in class. On the other hand, we will prove that this is tight as any graph containing at least $100n^{3/2}$ edges must also contain a $\mathcal{K}$: First, using a Lemma from class, we may find a subgraph with minimum degree at least $50n^{3/2}$. In that subgraph, there must be a C_4 due to the extremal number of C_4s, but also every vertex is of degree at least 4 and thus any C_4 is also contained in a $\mathcal{K}$.

Question 2.

a) Let $(S, V \setminus S)$ be a cut in the graph and assume that $vol(S) \leq vol(V \setminus S)$. Denote by $D := E(S, V \setminus S)$ the set of edges that cross this cut. Denote by $C_1, \ldots, C_r$ the connected components of $G[S]$, these are all distinct connected components of $G - D$ and each of them has volume at most $vol(S) \leq \frac{1}{2}vol(V)$. In particular, from our assumption,

$$vol(S) = \sum_{i=r}^{r} vol(C_r) \leq |D|/\varphi = |E(S, V \setminus S)|/\varphi.$$

We conclude that $|E(S, V \setminus S)|/vol(S) \geq \varphi$, and as this is true for every cut in the graph its conductance is at least φ.

b) If G is disconnected then denote by $S \subsetneq V$ a connected component of it. As we have seen in class (or by a simple calculation), both $x := 1_S$ and $y := 1_V$ are in the kernel of L_G. Observe that x, y are linearly independent as all of their coefficients are non-negative but x has zeroes where y does not. In particular, there exists a scalar α such that $(x - \alpha y) \perp \overrightarrow{d}$: simply pick $\alpha :=< x, \overrightarrow{d} > / < y, \overrightarrow{d} >$ (note that the denominator is not zero). Thus we found a non-zero vector orthogonal to $\overrightarrow{d}$ such that the quotient $\frac{z^t L_G z}{z^t D_G z} = 0$. As we have proved in class, $\lambda_2(N_G)$ is the minimum of this quotient over all such vectors and thus it is zero. We can prove the other direction together with Clause (c) as it is a stronger (quantitative) version of it.

c) Assume that G is connected. Let $(S, V \setminus S)$ be any cut in the graph with $vol(S) \leq vol(V \setminus S)$. As the graph is connected, $|E(S, V \setminus S)| \geq 1$, and as an undirected simple graph $vol(S) \leq \frac{1}{2}vol(V) = m \leq \frac{n^2}{2}$. We conclude that

$$\frac{|E(S, V \setminus S)|}{vol(S)} \geq \frac{2}{n^2}.$$

In particular, the conductance of G is at least $\varphi \geq 2/n^2$. Thus, by Cheeger's inequality as proven in class we have

$$\lambda_2(N_G) \geq \frac{1}{2}\varphi(G)^2 \geq \frac{1}{2}\left(\frac{2}{n^2}\right)^2 = \frac{2}{n^4}.$$

Question 3.

We show that Prim's algorithm only uses MinCut queries. Consider the following algorithm:

- Initialize $T \leftarrow \emptyset$ an empty tree and $S \leftarrow \{v\}$ for an arbitrary vertex $v \in V$.
- For $(n-1)$ steps do the following:
  - Ask a MinCut query for S, denote the answer by (u, w) with $u \in S, w \notin S$.
  - Update $T \leftarrow T \cup \{(u, w)\}$ and $S \leftarrow S \cup \{w\}$.
- Return T.

We can prove the resulting T is an MST by induction: For a graph of size 1 the statement is trivial. For a larger graph, consider the first edge e we add to T, it is the minimal in some cut (as is every edge returned by a MinCut query) and is thus in some MST as shown in class, the rest of the algorithm is equivalent to running the same algorithm on the graph G with the edge e contracted and its contracted vertex being the initial vertex in S. As seen in class, a contracted MST edge in addition to the MST of the contracted graph is an MST of the entire graph.

The algorithm makes only $(n-1)$ MinCut queries, which is also optimal (for example as this is the number of edges in any possible spanning tree and each query returns only a single edge).

Question 4.

A subset of the vertices spans a clique if and only if the same subset of vertices is an independent set in the complementary graph G^c (in which we flip the existence or non-existence of every edge). In particular, the minmum number of cliques needed to cover G equals the minimum number of independent sets needed to cover G^c which is simply its chromatic number (G^c). We can thus use the algorithm from class to compute this chromatic number.