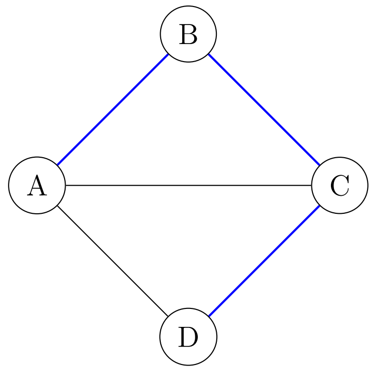
CPSC 320 2024W1: Assignment 5 Solutions

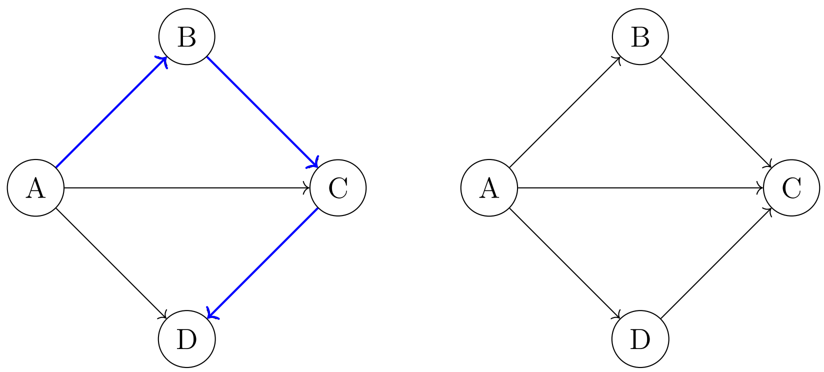
3 The Path To Victory 2.0

A Hamiltonian Path is a path in a graph that visits each vertex exactly once. In this question, we consider the variant of the Hamiltonian Path problem with the start and end node specified: that is, given a graph $G = (V, E)$ and two nodes s and t, does there exist a path from s to t that visits every node in G exactly once? You can may assume that G contains at least three vertices.

The Hamiltonian Path problem can be applied to graphs that are either undirected or directed. For example, the undirected graph below has a Hamiltonian Path from A to D given by (A, B, C, D), shown in blue:



For the directed graphs below, the graph on the left has a Hamiltonian Path from A to D, while the graph on the right does not have a Hamiltonian Path from A to D.



We refer to the undirected version of this problem as **UHP** and the directed version as **DHP**.

1. [3 points] Give a reduction from UHP to DHP.

   An edge (u, v) in an undirected graph is functionally equivalent to the two edges $(u, v), (v, u)$ in a directed graph. Specifically, if an undirected graph contains the edge (u, v), then a potential solution to UHP could have u immediately preceding v, or v immediately preceding u.

   Therefore, our reduction is: define a new graph G' by taking all the vertices in V, and replacing each undirected edge (u, v) in G with the directed edges $(u, v), (v, u)$ in G'. The nodes s and t in DHP are the same as the nodes s and t in UHP.

2. [4 points] Prove the correctness of your reduction from UHP to DHP. That is, prove that the answer to your reduced DHP instance is YES if and only if the answer to UHP is YES.

   Proof that YES to UHP -> YES to DHP: suppose the undirected graph G has a Hamiltonian Path $v_1, v_2, \ldots v_n$. Each successive pair of vertices (v_i, v_{i+1}) in this path is connected by an edge; this means in the reduced directed graph, there will be an edge (v_i, v_{i+1}) (and also (v_{i+1}, v_i)), because we added two directed edges for each directed edge. The graph G' in DHP has all the same vertices as the UHP graph G, so the Hamiltonian path in UHP is also a Hamiltonian path in DHP.

Proof that YES to DHP -> YES to UHP: suppose the DHP graph G' has a Hamiltonian path $v_1, v_2, \ldots v_n$. Between each successive pair of vertices v_1 and v_{i+1}, there is a directed edge (v_i, v_{i+1}). Our reduction would have only added this edge if the (undirected) edge (v_i, v_{i+1}) existed in G. Therefore, the same set of nodes $v_1, v_2, \ldots v_n$ also define a Hamiltonian Path in the UHP graph G.

3. [7 points] Give a reduction from DHP to UHP. You **do not** need to prove the correctness of this reduction, but you should clearly explain the key components of your reduction and why they are there.

   This direction is trickier because a directed edge (u, v) imposes a rule – namely, that u can immediately precede v in the Hamiltonian Path but not the other way around – that we can't directly encoded in UHP.

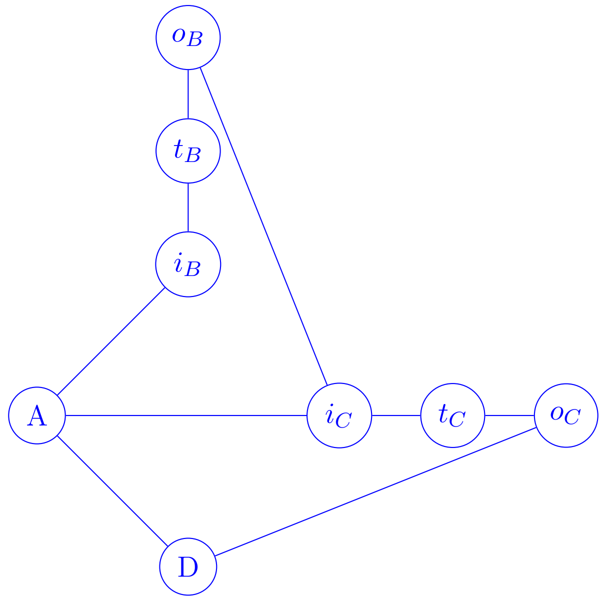
   To think about how we can express this rule in UHP "language", suppose a directed graph has a Hamiltonian Path $s \to v_2 \to \ldots \to v_{n-1} \to t$. We can say this as "exit v_1, enter v_2, exit v_2, …, enter v_{n-1}, exit v_{n-1}, enter t." Thus our solution needs to exit s, enter t, and enter and exit every other node exactly once. This sounds very much like a UHP problem if we treat each "enter" and "exit" as a separate node. Thus, we will encode this in an undirected graph by defining nodes that represent each "enter" (incoming edges) and "exit" (outgoing edges) from each node.

   With that formulation, however, there's nothing to prevent our reduction from mixing and matching the order of enters and exits. E.g., if I have nodes v_1 and v_2 both connected (so there's an edge (v_1, v_2) and an edge (v_2, v_1)) my reduction could theoretically choose to go from "enter v_1" to "exit v_2" to "enter v_2" to "exit v_1," which doesn't quite match what our Hamiltonian path should do, and could result in our reduction being able to construct a Hamiltonian path in the UHP graph where none exists in the DHP graph. To force our reduction to select the v_i enter and exit nodes together, we will also add a v_i "transit node" with edges defined from enter v_i, transit v_i, and exit v_i. This way, in order to include "transit v_i" on the path (and not visit one of its neighbours more than once), we will need to visit the nodes in that order (and our reduction can't "mix and match" the order of enter and exit nodes).

   Our complete reduction (given a DHP instance $G = (V, E)$ and nodes s and t) is:

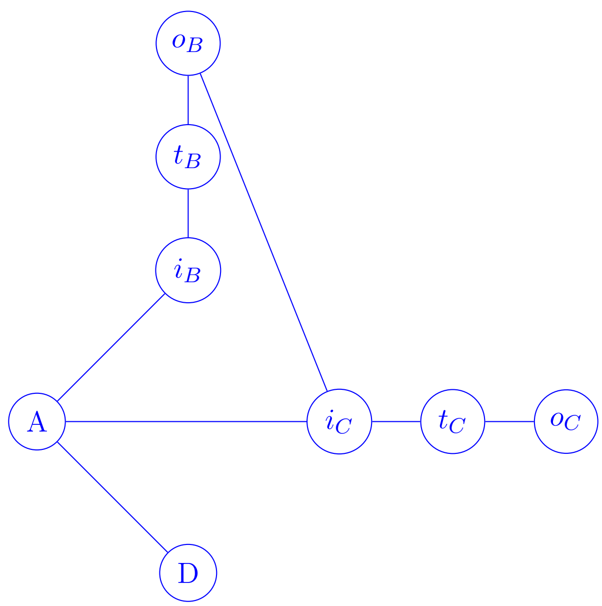
   - For the vertices s and t, define nodes s and t in G'.
   - For each vertex $v_j \in V$ not equal to s or t, define three nodes o_j, t_j and i_j with edges (o_j, t_j) and (t_j, i_j).
   - For each edge $(s, v_j) \in E$, define an edge from s to i_j. We'll ignore any edges pointing to s, since we know these won't be part of our directed Hamiltonian Path – and, more practically, we haven't defined an "incoming" node for s, so we can't include these edges without breaking our reduction.
   - For each edge $(v_j, t) \in E$, define an edge from o_j to t (again, we ignore any edges pointing from t).
   - If $(s, t) \in E$, we add (s, t) to G' (we could also choose not to do this because we know it won't be part of the Hamiltonian Path).
   - For each edge $(v_j, v_k) \in E$ (with $v_j, v_k \neq s, t$), define an edge from o_j to i_k.
   - The answer to DHP is YES if and only if the answer to UHP is YES.

   Since this reduction is a bit tricky to wrap your head around, we'll include some examples too. For the directed graph in the question description with the Hamiltonian Path from A to D, the corresponding graph for UHP with $s =$ A and $t =$ D is:



Notice that, as we would hope, this graph contains a Hamiltonian Path from A to D – namely (A, i_B, t_B, o_B, i_C, t_C, o_C, D).

On the other hand, for the directed graph in the question description that does not have a Hamiltonian Path, our reduction produces:



This UHP graph does **not** have a Hamiltonian Path from A to D.

4 Strategically Placed Krispy Kremes

This SPKK problem is also in Tutorial 6. UBC Rec student leaders are planning their next fundraiser, and are seeking your help in identifying strategic locations to set up their stands of Krispy Kremes. They have a map showing n locations of buildings and outdoor spots on campus. Their k stands need to be set up in the outdoor spots. They want you to select k spots such that the maximum distance from any of the n locations to a stand is as small as possible.

An instance of the Strategically Placed Krispy Kremes decision problem (SPKK) has a set V of size n, a subset S of V, an integer k, $1 \le k \le |S|$, and a symmetric matrix $d[1..n][1..n]$, plus an additional nonnegative integer b. The problem is to determine if there is a subset $S' \subseteq S$ of size k, such that

$$\max_{v \in V} \min_{s \in S'} \{d[v][s] \mid s \in S'\} \le b.$$

1. [8 points] Provide a reduction from the Dominating Set problem to SPKK. Briefly justify why the reduction is polynomial-time (one sentence is sufficient). Carefully explain why your reduction is correct (two paragraphs is sufficient).

   Let $I = (G, k)$ be an instance of Dominating Set, where $G = (V, E)$ and $V = \{1, 2, \ldots, n\}$. We construct an instance $I' = (V', S, d, k', b)$ of SPKK as follows. We set $V' = S = V$, so there are n locations and all are outdoor locations. Also,

   $$d[i][j] = \begin{cases} 0, & \text{if } i = j, \\ 1, & \text{if } (i, j) \text{ is in } E, \\ 2, & \text{otherwise.} \end{cases}$$

   We let $k' = \min\{k, n\}$ and let $b = 1$.

   To show correctness, we need to show that I is a Yes-instance of Dominating Set if and only if I' is a Yes-instance of SPKK.

   - First suppose that I is a Yes-instance of Dominating Set. Let S' be a set of size k such that every node of V is either in S' or connected to a node of S' by an edge of E. Then in instance I' of SPKK, the nodes of S' are outdoor locations, and every other location is within distance 1 of a node of S'. So I' is a Yes-instance of SPKK.
   - Now suppose that I' is a Yes-instance of SPKK. Then there is a subset S' of V' of size k such that all nodes of V' are within distance 1 of a node in S'. Since $V = V'$ and since pairs of locations that are of distance 1 from each other correspond exactly to pairs of nodes of instance I that are connected by an edge, it must be that S' is a dominating set of G. So I is a Yes-instance of Dominating Set.

   This completes the proof of correctness.

5 Very Special Problems

In this question, we consider special cases of NP-complete problems. Formally, we say that Problem B is a **special case** of Problem A if every instance of Problem B can be viewed as an instance of Problem A. We've seen examples of this in class: for example, 3-SAT is a special case of SAT (where every clause has length 3). Minimum spanning tree is a special case of Steiner Tree, in which the set of vertices we need to connect includes all the vertices in V.

1. [4 points] Consider the **Bounded-Leaf Spanning Tree Problem (BLST)**: given a graph $G = (V, E)$ and an integer k, does G have a spanning tree with no more than k leaves?

   Give an NP-complete problem that is a special case of BLST, and justify why this problem is a special case.

   The answer is **Hamiltonian Path** (...again!). (This is possibly not the only answer, but in our opinion it is definitely the most straightforward.) A spanning tree with two leaves is a "stick" (i.e., a path) that includes all nodes in the graph, which is a Hamiltonian Path by definition. Therefore, Hamiltonian Path is a special case of BLST with $k = 2$.

2. [3 points] You showed in the previous question that there is an NP-complete problem that is a special case of BLST, and it is not difficult to show that BLST is in NP (though we are not asking you to do this). Does this imply that BLST is NP-complete? Justify your answer.

   Yes, this does show that BLST is NP-complete. Recall that the NP-complete problems are the NP problems with the property that, if we can solve this problem in polynomial time, we can solve **any** problem in NP in polynomial time. If we could solve BLST in polynomial time, we could solve Hamiltonian Path in polynomial time (because Hamiltonian Path is a special case of BLST), and then in turn solve all problems in NP in polynomial time. (Additionally, you could also trivially reduce Hamiltonian Path to BLST, because Hamiltonian Path is already a BLST instance with $k = 2$.) Therefore, BLST is NP-complete.

3. [4 points] Give an example of a polytime-solvable problem you have seen in this class that is a special case of an NP-complete problem. Justify your answer.

   There are several correct answers here. Here are a few (this is just what I can come up with off the top of my head and is not an exhaustive list):

   - The volunteer meeting problem (A3) is a special case of set cover where all sets are contiguous (e.g., an interval corresponds to a set of all the discretized times between time i and time j).
   - Similarly, interval scheduling (textbook readings) is a special case of set packing, again with all sets being contiguous.
   - The shortest path problem (discussed as part of the DIAM worksheet) is a special case of Steiner Tree where exactly two nodes are shaded.
   - The Bipartite Graph problem (tutorial 1) is a special case of graph colouring with 2 colours. We can solve this in polynomial time using depth first search; see Kleinberg and Tardos, section 3.4.
   - 2-SAT is a special case of SAT that can be solved in polynomial time, though we did not mention in this course how to do this (and, for a 2-SAT answer to receive full credit for justification, we will require that you give some explanation here). Essentially, we can construct a 2-SAT instance as a directed implication graph, find all the strongly connected components in the graph, and we can satisfy the 2-SAT instance if and only if the vertex corresponding to x and the vertex corresponding to its negation $\overline{x}$ are in different connected components for all variables x. See details here.

   Note that Subset Sums mod M (A4) is **not** a correct answer for the same reason that Subset Sum is solvable only in pseudo-polynomial time (see your textbook readings): namely, the runtime depends on the numerical value of the target sum, which is not polynomial in the "size" of the input. This is the number of bits in the target sum; the actual value is exponential in this quantity.