# CPSC 320 2024W1: Assignment 5

Hansel Poe

5 December, 2024

This assignment is due **Friday, December 6 at 7 PM**. Late submissions will not be accepted. All the submission and formatting rules for Assignment 1 apply to this assignment as well.

## 1 List of names of group members (as listed on Canvas)

Provide the list here. This is worth 1 mark. Include student numbers as a secondary failsafe if you wish.

- Hansel Poe - 82673492

## 2 Statement on collaboration and use of resources

To develop good practices in doing homeworks, citing resources and acknowledging input from others, please complete the following. This question is worth 2 marks.

1. All group members have read and followed the guidelines for groupwork on assignments given in the Syllabus).

   ● Yes ○ No

2. We used the following resources (list books, online sources, etc. that you consulted):

3. One or more of us consulted with course staff during office hours.

   ● Yes ○ No

4. One or more of us collaborated with other CPSC 320 students; none of us took written notes during our consultations and we took at least a half-hour break afterwards.

   ○ Yes ● No

   If yes, please list their name(s) here:

5. One or more of us collaborated with or consulted others outside of CPSC 320; none of us took written notes during our consultations and we took at least a half-hour break afterwards.

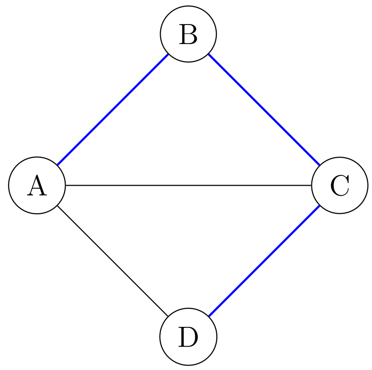
   ○ Yes ● No

   If yes, please list their name(s) here:

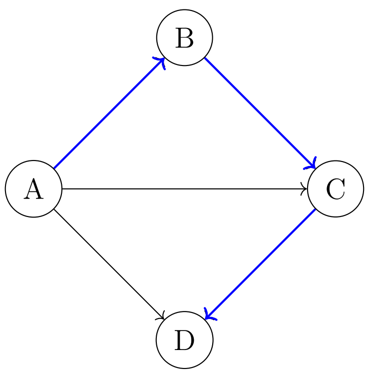
## 3 The Path To Victory 2.0

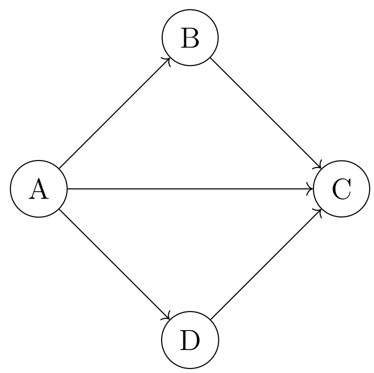
A Hamiltonian Path is a path in a graph that visits each vertex exactly once. In this question, we consider the variant of the Hamiltonian Path problem with the start and end node specified: that is, given a graph $G = (V, E)$ and two nodes $s$ and $t$, does there exist a path from $s$ to $t$ that visits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

   Instance transform algorithm:
   Let $I = (G = (V, E), s, t)$ be our undirected instance of the Hamiltonian problem and $I' = (G' = (V', E'), s', t')$ be the directed instance.

   First we can simply do an identity mapping of the starting and ending nodes, namely let $s' = s$ and $t' = t$. We also do the same for the vertices in our graph, $V' = V$.

   Now, for the edges, for any unordered pair $\{u, v\}$ in $E$, we add two ordered pairs $(u, v)$ and $(v, u)$ into $E'$.

   Solution transform algorithm:
   We simply map YES to $I'$ to YES to $I$ and NO to $I'$ to NO to $I$

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.

First we prove YES to $I \implies$ YES to $I'$. Assume $S$ is our solution set for UHP forming the hamiltonian path from $s$ to $t$. We will construct the solution $S'$ for DHP containing the path from $s'$ to $t'$.

Consider the first edge in our path $\{s, u\}$.We choose $(s, u)$ to be our first edge in $S'$.

Then, for any $\{a, b\}$ and $\{b, c\}$ in $S$, If we have $(a, b)$ in $S$, we include the edge $(b, c)$ in our $S'$. This ensures we do not have two arrows pointing to the same node in our $S'$. By doing this for every edge in $S$, we will have inductively constructed our path from $s'$ to $t'$.

Now, we prove YES to $I' \implies$ YES to $I$. Now we assume we have the solution set $S'$ for DHP and try to construct $S$ for UHP.

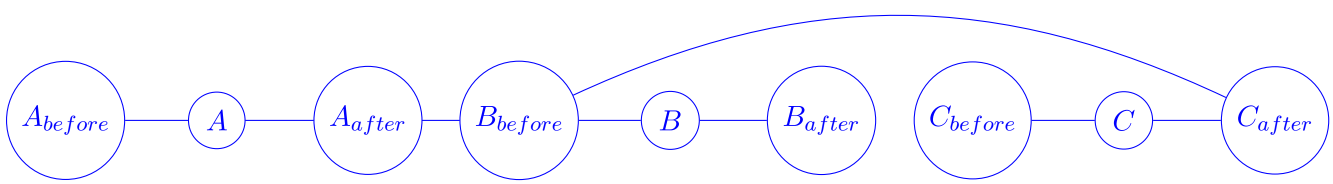
For any edge $(a, b)$ in $S'$, we simply include $\{a, b\}$ in $S$. Doing this indcutively for all edges in $S'$, we will have constructed our path from $s$ to $t$.

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.

   Instance transform algorithm:
   Let $I = (G = (V, E), s, t)$ be our DHP instance and $I' = (G' = (V', E'), s', t')$ be our UHP.

   For each vertex $v \in V$, we add three nodes $v_{before}, v, v_{after}$ to $V'$ and we add the edges $\{v_{before}, v\}$ and $\{v, v_{after}\}$ into $E'$.

   Next, for any $(a, b) \in E$, we add $\{a_{after}, b_{before}\}$ into $E'$. This will ensure that if we have two arrows pointing to the same node, we "break" our path. For instance, if we have $A \to B \leftarrow C$. Our graph will look like:




   Finally, for the starting and end points, we simply let $s' = s$ and $t' = t$

   Solution transform algorithm: We simply map YES to $I'$ to YES to $I$ and NO to $I'$ to NO to $I$

## 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).

Instance transform algorithm:
Let $I = (G = (V, E), K)$ be an instance of Dominating Set, we will map this to an instance $I' = (V', S, k, d[1 \ldots n][1 \ldots n], b)$ of SPKK.

We start by simply mapping the vertices of DS to set of locations of SPKK, that is $V' = V$ .Then we also set the outside locations $S$ to $V = V'$, this is because in our DS, we can choose any vertices to include in our solution. In SPKK this translates to having our possible stand locations (outdoor locations) be all locations.

Next, we let $k = K$, the number of stands we have corresponds to the number of maximum vertices allowed in our solution set to DS. This ensures that our stands (chosen vertices) is at most $K$

Finally, we let the distance $d[u][v]$ be the shortest path from vertex $u$ to $v$ in our DS graph and $b = 1$. if no path exists from $u$ to $v$, we set the distance to $\infty$ and a distance from a node to itself is $0$ .This whole setup works because in a correct solution to DS, the shortest path from any vertex to a solution vertex is less than or equal to one (For any $v$ outside of our solution, there is an edge connecting $v$ to any node in our solution), which corresponds to SPKK's requirement that the maximum distance between any location to a stand is less than or equal to $b = 1$

Transform Solution Algorithm: We simply map YES to $I'$ to YES to $I$ and NO to $I'$ to NO to $I$

To show that the transformation is poly-time, we first note that mapping the solution algorithm takes constant time, Mapping $S$ and $V'$ to $V$ also takes constant time (we just use the same array or perhaps make a copy in linear time).Setting $k = K$ and $b = 1$ are also constant.

Finally, to construct the distance matrix, we can use the modified BFS from A2(polytime) on all nodes, which overall can be done in poly-time. Thus, our transformation is bounded poly-time.

## 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.

   Hamiltonian path problem is a special case of BLST, although we won't specify a start and end point like in (3). This is because a path which visits every node is a spanning tree with 2 leaves (the start and end node) and so the hamiltonian path problem is simply the 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! Because all instances of the Hamiltonian path problem can be framed as BLST, it must follow that we can do a reduction from Hamiltonian Path Problem to BLST. Meaning BLST is at least as hard as HPP, an NP-Complete problem. This along with the fact that BLST is in NP implies that 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.

Dominating set problem is a special case of the SPKK problem, where:

- $b = 1$
- $V =$ set of vertices of G in DS
- $S =$set of vertices of G in DS
- $k = K$ in DS
- $d[u][v] =$ shortest path between $u$ and $v$ in G in DS

The full reduction is given in (4). Because the Dominating Set is an NP-Complete problem and we can reduce it to SPKK, it must follow that SPKK is NP-Complete.