Exercise 2a

Definition 1 (2-directed Hamiltonicity)**.** Given a graph $G = (V, E)$, G is said to be 2-directed Hamiltonic, iff there exist 2 simple disjoint circuits in the graph with at least 3 nodes each, where the union of both circuits is V.

The decision problem ($2DHAM$) is then: given a graph G, is it 2-directed Hamiltonic?

Proof. We are asked to prove that 2-directed Hamiltonicity ($2DHAM$) is in $\mathcal{NPC}$. Let us assume that directed Hamiltonian cycle (DHC) is in $\mathcal{NPC}$[1].

The proof has two steps.

- I need to prove that $2DHAM$ is in $\mathcal{NP}$.
- I need to prove that some $\mathcal{NPC}$ problem X can be reduced in polynomial time to $2DHAM$ (i.e. $X \propto 2DHAM$ in polynomial time).

First of all, $2DHAM$ is trivially in $\mathcal{NP}$. Given a certificate for an instance of $2DHAM$, i.e. two simple directed circuits in $G = (V, E)$, we can check that each of the circuits has at least 3 nodes (that is trivially in $\mathcal{P}$), that each of the circuits is simple (again trivially in $O(|V|)$ time and space by traversing each circuit and using a hash table), that the union of both circuits encompasses V (again trivially in $O(|V|)$ time and space with a help of a hash table and a simple traversal). Therefore, $2DHAM \in \mathcal{NP}$.

For the second step of the proof I'll use a reduction from DHC. Given an instance of DHC, i.e. a directed graph $G = (V, E)$ and the question "Does G have a directed Hamiltonian circuit?", we can transform this to an instance of $2DHAM$ in the following fashion. Let $G' = (V', E')$ be constructed thus.

$$\begin{aligned} V' &= V \cup \{x_1, x_2, x_3\} \\ E' &= E \cup \{(x_1, x_2), (x_2, x_3), (x_3, x_1)\} \end{aligned}$$

I.e. I augment G with a directed Hamiltonian circuit x_1, x_2, x_3 disconnected from the rest of the graph. This transformation is obviously polynomial in $|V|$ and $|E|$. Now, G' is 2-directed Hamiltonian if and only if G has a Hamiltonian circuit. Why?

- If G has a Hamiltonian circuit, then G' has the following 2 Hamiltonian circuits: the directed Hamiltonian circuit of G spanning V (this is the assumption) plus an additional circuit $\{x_1, x_2, x_3\}$. The latter is trivially Hamiltonian. Thus G' has 2 directed Hamiltonian circuits and the union of the nodes participating in both covers the entire V'.
- Conversely, assume that G' has 2 directed Hamiltonian circuits. One of these is the (artifically) constructed $\{x_1, x_2, x_3\}$. Since both circuits span the entire V', it means that the second directed circuit must span the entire $V' \setminus \{x_1, x_2, x_3\} = V$. But a simple directed circuit spanning V will be exactly a directed Hamiltonian circuit for $G = (V, E)$.

[1] We are allowed to assume that (undirected) Hamiltonian cycle problem is in $\mathcal{NPC}$. However, I'll demonstrate later that the directed version, DHC, is in $\mathcal{NPC}$ as well. Since it is ultimately irrelevant which $\mathcal{NPC}$ problem is used in the reduction, and I prefer DHC.

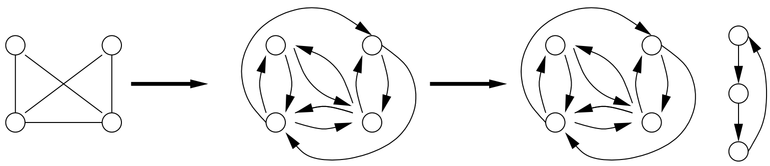

Figure 1: Reductions for NP-completness proof for 2a

Finally, there exist 2 trivial cases which have not been covered yet: G could have 1 or 2 nodes. Since there are 3 graphs in total possible with this number of nodes, for each of them the transformation can use a pre-programmed answer (a 1-node graph is defined to have a Hamiltonian circuit; a 2-node graph is defined NOT to have such a circuit. The transformation uses this definition to adjust the answers).

So, $DHC \propto 2DHAM$ (just proven) and $DHC \in \mathcal{NPC}$ (assumed). These imply that $2DHAM \in \mathcal{NPC}$. □

The last step is to demonstrate that $DHC \in \mathcal{NPC}$. We could assume without proof that (undirected) Hamiltonian circuit, HC, is in $\mathcal{NPC}$. DHC is obviously in $\mathcal{NPC}$ by using the following transformation from HC - each undirected edge (x_1, x_2) in the given HC-instance is mapped to two directed edges $(x_1, x_2), (x_2, x_1)$. The existence of a Hamiltonian circuit in one version obviously implies the existence of the Hamiltonian circuit in the other version (we follow the same path, which is always possible since an undirected edge results in directed edges going in both directions). $HC \propto DHC$ is obviously polynomial (DHC has twice as many edges but is otherwise identical).

The reductions for the example graph will proceed as outlined in figure 1.

Exercise 2b

Definition 2 (NP-hard)**.** Any problem Π, whether a member of $\mathcal{NP}$ or not, to which we can reduce an $\mathcal{NPC}$ problem. Π may be a function problem or a decision problem.

For the purpose of this exercise we are permitted to assume that "hamiltonicity" is in $\mathcal{NPC}$. I'll assume that both Hamiltonian cycle (HC) and Hamiltonian path (HP) are in $\mathcal{NPC}$.

(i) X: Find the length of the longest simple path between x and y. $\mathbf{X} \in \mathcal{NPH}$ (however, $X \notin \mathcal{NPC}$). Thus, X is "genuinely" NP-hard.

The fact that $X \in \mathcal{NPH}$ is proved by finding a suitable reduction from an $\mathcal{NPC}$ problem. An obvious choice is HP. Assuming that we can solve X, $HP \propto X$ in the following fashion. Given a graph $G = (V, E)$ and given a Turing machine M that solves X, G has a Hamiltonian path iff M finds a longest simple path of length $|V| - 1$ for a given node x and some other node $y \in V$. So, $HP \propto X$. This proves that $X \in \mathcal{NPH}$, but I still need to prove that $X \notin \mathcal{NP}$. This is by the fact that X is not a decision problem.

(ii) X: Find the length of the shortest simple path between x and y. $\mathbf{X}$ is polynomial.

And, I might add, obviously so, since Dijkstra's algorithm solves X in time polynomial in graph's size. Given a problem instance for $X, G = (V, E)$ assign a weight of $+1$ to each edge and run Dijkstra's algorithm from x. The length of the path returned by the algorithm *is* in fact the shortest path, if it at all exists. The path is obviously simple, since visiting the same node twice implies a loop which is impossible since it would have been removed given how Dijkstra's algorithm works and given that all the edges are $+1$ in weight.

(iii) X: Decide if there is a simple path between x and y that contains at least half the nodes (including x and y). $\mathbf{X} \in \mathcal{NPC}$.

The problem is obviously in $\mathcal{NP}$ (given a path between x and y, it is trivial to check in polynomial time that the path is simple and that its length is $\geq |V|/2$). Now for the reduction. I'll demonstrate that $HP \propto X$. Given an instance of HP, $G = (V, E)$ construct $G' = (V', E')$ in the following fashion. $V' = V \cup \{x_1, \dots, x_{|V|}\}$, $E' = E$. I.e. we augment the original graph with twice the number of nodes and keep the original edges. G' has a simple path between x and y iff G has that path (the x_is are not connected and cannot contribute to the path). If the length of that path is $|V'|/2$, it means that G has a simple path of length $|V|$. If the length of that path is less than $|V'|/2 = |V|$, the path in G is not Hamiltonian. This concludes the reduction.

So, $X \in \mathcal{NP}$ and $HP \propto X$. Since $HP \in \mathcal{NPC}$ (assumed), $X \in \mathcal{NPC}$.

(iv) X: Find a longest simple path between x and y in G. The algorithm is to return the resulting node sequence for one of the longest simple paths. $\mathbf{X} \in \mathcal{NPH}$.

This is not a decision problem, much like in (i). There is a trivial reduction though, $HP \propto X$ (HP exists iff the length of the longest path is exactly $|V| - 1$)

(v) X: Is there a clique in G containing x, y and 3 more nodes? If so, produce such a clique. $\mathbf{X} \in \mathcal{P}$.

The easiest way is to produce an algorithm. The critical part here is the fact that we have 2 starting nodes and need only 3 more (this is what distinguishes this problem from maximum clique or generalized clique's existence problems). The easiest is simply to sketch is a straightforward brute-force algorithm 1 on the next page.

Each step of the algorithm is obviously polynomial in $|V|$. Calculating neighbours of a node is linear in $|V|$. Calculating a set intersection where each set is bounded by $|V|$ is polynomial in $|V|$. All steps are polynomial, the algorithm is polynomial and therefore $X \in \mathcal{P}$.

```
procedure common(x_1, ..., x_k)
tmp ← neighbours(x_1)
for all x in x_2 ... x_k do
  tmp ← tmp ∩ neighbours(x)
end for
return  tmp {set of nodes that are connected to all nodes x_1 ... x_k}
end procedure
procedure cliqueness(x, y)
for all n_1 in common(x,y) do
  for all n_2 in common(x,y,n_1) do
    for all n_3 in common(x,y,,n_1,n_2) do
      return  x, y, n_1, n_2, n_3 {that's our clique}
    end for
  end for
end for
return  there is no clique
end procedure
```

Algorithm 1: Clique calculation for problem (v)