Homework 10

CS41, Spring 2023 Alina Palacios and Nina Zhuo

1. We save a copy of the graph G into G' such that $G' = G$. For each edge $e \in E$, we call CheckHam on $G' \backslash e$. If CheckHam returns true indicating that there exists a Hamiltonian cycle in $G' \backslash e$, we know that e is not a necessary edge in any one Hamiltonian cycle in G so we set $G' = G' \backslash e$. If CheckHam returns false, we know that e is necessary edge in the Hamiltonian cycle in G, so we keep e in G'. After iterating through each edge, we will be left with a G' that only has the edges necessary in a Hamiltonian cycle in G if one exists. We verify by calling CheckHam on G', returning G' if successful, false otherwise. We call CheckHam $O(|E|)$ times and CheckHam is an $O(n^k)$ time algorithm where $n = |V|$. Because G is a digraph, we know that $|E| \leq n^2$ such that total running time will be $O(n^2 n^k)$ or $O(n^\ell)$ where $\ell = k + 2$.

2. We know a cycle on a graph with n vertices will have at most n edges. Since each edge's weight is at most 2^n, the maximum weight of any one cycle on G will be $n2^n$.

   A naive approach would call BudgetTSP at most $n2^n$ times, incrementing the budget $b = 1$ up to $n2^n$ to find the lightest Hamiltonian cycle. Recognize that we can instead take a binary search approach to all possible values of b so we only call BudgetTSP at most $log(n2^n)$ times. We initialize b to be the median of all possible total weights of a Hamiltonian cycle in G, so $k = n2^n/2$.

   Case 1: If BudgetTSP returns false, we know that if there exists a Hamiltonian cycle in G, its total weight exceeds our current budget. Thus, we set b to be the median of all possible total weights from b to $n2^n$. We run BudgetTSP on our new b and repeat until either $b = n2^n$ indicating there is no Hamiltonian cycle in G or BudgetTSP does not return false.

   Case 2: If BudgetTSP does not return false, we know that there exists a Hamiltonian cycle in G with a total weight of $\leq b$. Because we want the lightest possible Hamiltonian cycle, we set b to be the median of all possible total weights from 0 to b.
   We run BudgetTSP on our new b and, and repeat until BudgetTSP returns false, indicating that there are no lighter cycles in G.

   We call BudgetTSP at most $O(log(n2^n))$ times such that the running time of our algorithm will be $O(log(n2^n) \cdot (nlogb)^k$ and we know that our budget b will never exceed $n2^n$. We solve to check that our algorithm runs in polynomial time.

$$
\begin{aligned}
log(n2^n) \cdot (nlog(n2^n))^k) &= log(n2^n) \cdot n^k log(n2^n)^k \\
&= log(n2^n) \cdot n^k log(n2^n)^k \\
&= log(n2^n) \cdot n^k k(log2^n)^k \\
&= n^k(logn + log(2^n))^{k+1} \\
&= n^k(logn + nlog(2))^{k+1} \\
&= n^k(logn + n)^k + 1 \\
&= n^k(logn + n)^{k+1} && \text{Observe that } logn + n \leq 2n
\end{aligned}
$$

$$= n^k(2n)^{k+1}$$
$$= n^k n^{k+1} 2^{k+1}$$
$$= n^{2k+1} 2^{k+1}$$

Constants are not considered in asymptotic running time analysis, so our algorithm runs in $O(n^m)$ where $m = 2k + 1$ which is polynomial time.

3. We will prove that HAM-PATH $\leq p$ BDD-SPAN by showing that a HAM-PATH can be reduced to at most a 2-k spanning tree.

   Consider a path in our undirected graph G. We know that the in-degree of the start vertex s is at most 1. The intermediate vertex u will have an in-degree of at most 2, since only one edge will go into u and at most one edge will go out from it. The end vertex t will have an in-degree of at most 1. Since we know that there are no cycles in the graph, it is not possible to ever go back and visit the same vertex more than once. Thus, we can conclude that all Hamiltonian paths are 2-spanning trees.

   With this information, we can show that HAM-PATH(G) $\leq p$ BDD-SPAN(f(G)), such that the output of HAM-PATH when inputting a graph G is the same as running BDD-SPAN on the output of HAM-PATH. We know that because all Hamiltonian paths are 2-spanning trees, that BDD-SPAN(f(G)) would return the same graph G and $k = 2$, thus HAM-PATH $\leq p$ BDD-SPAN.