CS120: Intro. to Algorithms and their Limitations	Anshu & Hesterberg
Sender–Receiver Exercise 4: Reading for Receivers	
Harvard SEAS - Fall 2023	*2023-11-02*

The goals of this exercise are:

- to develop your skills at understanding, distilling, and communicating proofs and the conceptual ideas in them,
- to practice reductions to SAT, and in particular how logic is useful for modelling problems

To prepare for this exercise as a receiver, you should try to understand the theorem statement and definition in Section 1 below, and review the material on Logic and Satisifiability covered in class on October 26 and October 31. Your partner sender will communicate the proof of Theorem 1.1.

1 The Result

In class, we saw how the (seemingly hard) problem of Graph k-Coloring can be efficiently reduced to CNF-Satisfiability (SAT). Although SAT also seems to be a hard problem (as we'll formalize in the last part of the course), this allowed all the effort put into SAT Solvers to solve many large k-coloring instances in practice.

In this exercise, you'll see a similar reduction for the *Longest Path* problem.

Input	: A digraph $G = (V, E)$ and two vertices $s, t \in V$
Output	: A *longest path* from s to t in G, if one exists

Computational Problem LongestPath

(Recall that a *path* is a walk with no repeated vertices.)

Actually, it will be more convenient to consider a version where the desired path length is specified in the input.

Input	: A digraph $G = (V, E)$, two vertices $s, t \in V$, and a path-length $k \in \mathbb{N}$
Output	: A path from s to t in G of length k, if one exists

Computational Problem LongPath

If we have an efficient algorithm for LongPath, then we can solve LongestPath by trying $k = n, n-1, \ldots, 0$ until we succeed in finding a path. The $k = n$ case is essentially the same as the *Hamiltonian Path* problem,[1] which is a special case of the notorious *Travelling Salesperson Problem (TSP)*. In the TSP, we have a salesperson who wishes to visit n cities (to sell their goods) in the shortest travel time possible. If we model the possible travel between cities as a directed graph, then Hamiltonian Path corresponds to the special case where all pairs u, v of cities either have a travel time of 1 (edge (u, v) present) or a very large travel time (edge (u, v) not present). In such a case, the only way to visit all cities in travel time at most $n - 1$ is via a Hamiltonian Path.[2]

[1] In the HamiltonianPath problem, we don't specify the start and end vertex; any path of length n suffices. But the two problems can be efficiently reduced to each other (exercise).

[2] Often in the TSP, it is also required that the salesperson return back to their starting city s. If we add edges of travel time 1 from all cities to the starting city, then we see that Hamiltonian Path is also a special case of this variant of the TSP.

The reduction from LongPath to SAT is given as follows.

Theorem 1.1. *LongPath on a digraph with n vertices, m edges, and a path length k reduces to SAT in time $O(n^2k)$.*

2 The Proof

Constructing a SAT instance φ from a LongPath instance (G, s, t, k).

Converting a satisfying assignment α to φ into a LongPath solution P.

Correctness of the Reduction.

Runtime of the Reduction.