Jonathan Tso
CS 301
HW 6

1. We know that the set P is decidable in polynomial time. That is, anything within the set P can be represented by a turing machine, TM N.

Consider if we make the transform of TM N, where we flip the accept and reject at the end of the TM. This can be done in polynomial time, as we have a finite number of accept and reject results under P. Thus, the transformation is done in P runtime, and so we can always create the complement of any TM N. Because the transformation is also P, the complement is closed for the set P.

2. Consider that the graph G has a start vertex S. With an arbitrary N number of nodes, we have a maximum of approximately N^N edges. Every additional node would be N+1 additional edges. This is an exponential runtime for increasing n. However, if we are looking for a graph with at least 2 paths to t and are provided 2 paths, we can verify if TWOPATH ∈ NP, with a verifier algorithm, X(TM)

1=2=3

TM X:

1. On one tape, write down the 2 provided strings of pathways in order.
2. From S, run through the first given string. If the end is t, go to 3. Else, reject.
3. From S, run through 2nd given string. If end is t, accept. Else, reject.

Our verifier will be in P runtime, because each string will be finite and at most N (# nodes). Since we have an exponential problem but a polynomial verifier, we can conclude TWOPATH is NP.

3. Consider that SORT = $\{\langle L, K \rangle \mid$ L is a list of elem and K is largest$\}$

We can prove this is $\in P$ by generating an algorithm that will decide it, with TM V.

TM V: Write the value of K on a second tape, and write L on the first tape.

From the head of the tape, look at the first element. If it is less than K, repeat this step, until you reach a blank. If you reach a blank, accept.

If the value on the 1st tape $>$ 2nd tape, K is not the largest, and reject.

Because the list L will be finite, we can do comparisons to K with n comparisons, which is polynomial in runtime.

Here, we have shown the setup takes at most $n+1$ and the actual comparison takes n runtime. Thus, this is $\in P$.