To do this quiz, you must have signed up by noon on 10/24 for one of the slots on the spreadsheet whose URL was BB-emailed on 10/23. And then you must show up DURING that slot. And then you (and whoever, if anyone, is sharing your slot) must spend that slot working (perhaps with hints from the TA) to prove (or substantially make progress on proving) that Few ⊆ ⊕P.

You CAN start your work on proving that earlier, joint with your (if any) time-slot partner or partners. However, it is cheating to discuss the quiz with anyone other than your time-slot partners before the quiz completes on November 1. (That does not put people in single-person times at a disadvantage, as just doing this quiz gets you a 100%.)

Even if you and your partner(s), if any, don’t work on proving that result before your time slot, I’d urge you to at least read the definitions of those two classes, as the definition of Few takes a moment to grasp. (But putting in time before your time slot is allowed and encouraged. You might even come in with a proof of this already. The proof is not utterly easy to find, though once one has it it is clear, and it sort of is what one would (or might) get to by going logically forward in approaching this problem.

Let me present to you those definitions. These def’s are from, as to the way they are presented, the paper:


but as that paper PROVES the theorem you are proving, you should not look at it until after you have taken the quiz. It is cheating to read the proof from that paper of the result the quiz is on, until after you have taken the quiz; you after taking the quiz do not have to wait until 11/1 to read the paper, however, if you’d like to read the paper.

By the way, as mentioned above, you don’t really have any incentive anyway to cheat and read the proof, since for this quiz, if you sign up for a slot by the deadline and show up for it and work with the TA on this for the period you are signed up for, you will get a 100%. That is, this quiz is a chance to get hands-on work done, and feedback, and guidance, rather than being a directly evaluative quiz.

And on we go to the definitions. Again, this is taken from the above paper’s presentation of them.

The class ⊕P, parity polynomial time, was defined and studied by Papadimitriou and Zachos 1983, and by Goldschlager and Parberry 1986, as a “moderate version” of Valiant’s counting class #P.

Definition [PZ83,GP86] ⊕P = \{L | there is a nondeterministic polynomial-time Turing machine N such that x ∈ L if and only if N(x) has an odd number of accepting paths\}.

... Few... captures the general notion of polynomial-path nondeterminism.

Definition[Long-Selman-1986] Let N be a nondeterministic Turing machine. \(\text{count}_N(x)\) is defined as the number of accepting paths of N on input x.
**Definition** Few is the class of all languages $L$ such that there is a nondeterministic polynomial-time Turing machine $N$, a polynomial-time computable predicate $Q(\cdot, \cdot)$, and a polynomial $q(\cdot)$, such that:

1. $x \in L$ if and only if $Q(x, \text{count}_N(x))$ and
2. $(\forall x)[\text{count}_N(x) \leq q(|x|)].$

In words, a language $L$ is in Few if there is a nondeterministic polynomial-time Turing machine $N$ that never has many accepting paths, and a polynomial-time computable predicate $Q$, such that on each input $x$, $Q$ can look at $x$ and the number of accepting paths of $N(x)$ and determine if $x \in L$.

Finally, let me mention now that a nice corollary of this result, the fact that the class called FewP (which is just NP except only regarding machines that have a polynomial bound on their numbers of accepting paths) is a subset of $\oplus P$, can be proven directly via a simpler and very cute proof. After you finish the quiz, you might as a challenge try to find that cute, simple, direct proof.

Have the TA you do this quiz with write here their signature and what time you did it and what your grade was, and they should before you even leave the room if possible put your grade on this into the 10/25 take-home quiz column in BB: