The gap theorem

(Adapted from Papadimitriou’s “Computational Complexity”)

THEOREM. There exists a computable function f: N →N, such that TIME(f(n))=TIME(2f(n)).

PROOF. Let M0, M1,… be a lexicographically ordered list of all TMs. For a natural number n, define a sequence with k1=2n, and . Define f(n) to be the smallest value k in this sequence where no input of length n causes any machine in M0, M1,…,Mn to have a runtime in [k,2k].

Now say that a language L is decided by a TM which runs in at most c2f(n) time for inputs of length n. Then there must exist another TM, say, Mj in our enumeration, which decides L with a runtime of at most 2f(n) for inputs of length n. For inputs of length n at least j, Mj must then run in time less than f(n) because of the way f was constructed. We modify Mj with a preprocessing stage with a runtime of O(n) which responds to all inputs of length less than j according to L to conclude that L is in TIME(f(n)).