Distributed Algorithms - Homework 1

1. Give a definition for the time complexity of an asynchronous algorithm: synchronous time, longest-chain time, bounded delay time. Give example(s) that show that these three definitions are distinct.

2. Prove the correctness of Dijkstra’s 1st algorithm under a centralized scheduler using a potential function argument.

3. Estimate the number of steps that it takes the system to self-stabilize in the worst case, for the following cases:
   - Dijkstra’s 1st algorithm with k=n under a centralized scheduler
   - Dijkstra’s 1st algorithm with k=n under a distributed scheduler

   Note: In each of these cases, give a function $f(n)$ which is as bad as possible, and show a family of initial configurations with $n$ processors which takes $g(n)$ steps to stabilize, for infinitely many $n$’s. Give functions $f(n)$ and $g(n)$ which are as close as possible to each other.

4. The question relates to the Burns-Pachl algorithm for self-stabilization.
   Prove or disprove: 
   The algorithm works correctly when the following conditions are removed from Rule A (one at a time):
   - a) $L_i \neq 0$
   - b) $T_{i-1} = 0$
   - c) $T_{i-1} \neq L_i - L_{i-1}$
   - d) $T_{i-1} < T_i$

   Note: Whenever you disprove, give a counter example (an execution) which is as short as possible.