Problem 1. (Exercise 1-3 from Minicourse on Multithreaded Programming) Prove that a greedy scheduler achieves the stronger bound:

\[ T_P \leq \frac{(T_1 - T_\infty)}{P} + T_\infty. \]  

(1)

Problem 2. (Exercise 1-6 from Minicourse on Multithreaded Programming) Professor Tweed takes some measurements of his (deterministic) multithreaded program, which is scheduled using a greedy scheduler, and finds that \( T_1 = 80 \) seconds and \( T_{64} = 10 \) seconds. What is the fastest that the professor’s computation could possibly run on 10 processors?

Problem 3. (Different Speed Processors) In this problem we consider how to schedule on processors of different speeds. Let there be \( p \) processors \( 1 \ldots p \), where processor \( i \) has speed \( \pi_i \) steps/time. Assume that \( \pi_1 \geq \pi_2 \geq \cdots \geq \pi_p \). Let \( W_1 \) represent the total work, that is the total number of nodes in the dag \( G \). Let \( W_\infty \) represent the critical path length of the graph, that is, the number of nodes in the longest chain in \( G \). Let \( \pi_{\text{ave}} \) steps/time be the average speed of the processors, that is, \( \pi_{\text{ave}} = \sum_{i=1}^{p} \pi_i/p \). Let \( T_p \) represent the optimal time to execute \( G \) on \( p \) processors.

Problem 3.a. Briefly explain why an arbitrary greedy schedule may perform poorly.

Problem 3.b. Describe a good greedy scheduler for this system.

Problem 3.c. Prove an analog of Graham/Brent for your scheduler. You should be able to show that:

\[ T_P \leq \frac{W_1}{p \, \pi_{\text{ave}}} + \left( \frac{p-1}{p} \right) \frac{W_\infty}{\pi_{\text{ave}}}. \]  

(2)

Problem 3.d. Can you still show that the time to complete all tasks is within a factor of 2 of optimal? Why or why not?