Project 2 (35 pt.)

This project is individual work, and each student must solve it independently.

Canvas submission instruction:
Step (1). In your first submission attempt, submit a **zip** file of your source codes and executable file;
Step (2). In your second submission attempt, submit a **pdf** project report.
If you would like to update any part (codes in zip or report in pdf) of your submission, redo the above two steps, such that the most recent submission attempt is always your pdf report and the second most recent submission attempt is always your zipped codes.

Instructions on project report preparation:

1. Use PowerPoint to prepare your report and submit as pdf, because you will need to put together pictures, screenshots, and texts in your report.
2. Test your codes with 5 input examples (an input example can be described by one task graph and one corresponding execution time table, other setups such as T_send, T_cloud, T_receive, number of local cores, and their power consumption P1, P2, and P3 can be kept the same):
   (1) the exact same one as Figure 1 of the paper;
   (2) the same number of tasks as in Figure 1 of the paper, but change connections in the task graph (in this case, the execution time table in Figure 1 can be used directly);
   (3) increase task number to 20, redesign the task graph (in this case, you need to add 10 more rows in the execution time table in Figure 1, and make sure that core 1 is the slowest and core 3 is the fastest in the added 10 rows);
   (4) base on the previous input example (3), redesign its task graph by using multiple entry tasks (in this case, the execution time table in input example (3) can be used directly)
   (5) base on the previous input example (3), redesign its task graph by using multiple entry tasks and multiple exit tasks (in this case, the execution time table in input example (3) can be used directly)
3. For each input example, write in your report: (a) figure of the input example including its task graph and its execution time table; (b) your program output in screenshot of initial scheduling, visualized initial scheduling result (as Figure 3 in the paper), report of total energy consumption of initial scheduling from your program, and manually calculate the total energy consumption of initial scheduling showing the derivation process such as for E1, E2, E3, E_cloud; (c) your program output in screenshot of final scheduling, visualized final scheduling result (as Figure 4 in the paper), report of total energy consumption of final scheduling from your program, and manually calculate the total energy consumption of final scheduling showing the derivation process such as for E1, E2, E3, E_cloud; and (d) a summary of T_total of initial scheduling, T_total of final scheduling, E_total of initial scheduling, E_total of final scheduling.
4. For input example (1), if your initial scheduling result is different from the paper, analysis the reason (maybe by demonstrating task priorities)

<u>Problem Description:</u>

Implement and test the proposed algorithms in the “Task Scheduling” paper. You can use similar set up as in the paper. You can set up T_max in task migration around 1.5X of T_total of initial scheduling.