Algorithm Design and Analysis
Assignment 3

1. (25 points) Give a linear-time algorithm that takes as input a tree and determines whether it has a *perfect matching*: a set of edges that touch each node exactly once.

2. (25 points) Consider you are a driver, and you plan to take highways from A to B with distance D. Since your car's tank capacity C is limited, you need to refuel your car at the gas station on the way. We are given n gas stations with surplus supply, they are located on a line together with A and B. The i-th gas station is located at d_i that means the distance from A to the station, and its price is p_i for each unit of gas, each unit of gas exactly support one unit of distance. The car's tank is empty at the beginning and so you can assume there is a gas station at A. Design efficient algorithms for the following tasks.

   (a) (5 points) Determine whether it is possible to reach B from A.

   (b) (20 points) Minimized the gas cost for reaching B.

3. (25 points) The problem is concerned with scheduling a set J of jobs $j_1, j_2, ..., j_n$ on a single processor. In advance (at time 0) we are given the earliest possible start time s_i, the required processing time p_i and deadline d_i of each job j_i. Note that $s_i + p_i \leq d_i$. It is assumed that s_i, p_i and d_i are all integers. A schedule of these jobs defines which job to run on the processor over the time line, and it must satisfy the constraints that each job j_i starts at or after s_i, the total time allocated for j_i is exactly p_i, and the job finishes no later than d_i. When such a schedule exists, the set of jobs is said to be feasible. We allow a preemptive schedule, i.e., a job when running can be preempted at any time and later resumed at the point of preemption. For example, suppose a job j_i has start time 6, processing time 5 and deadline 990, we can schedule j_i in one interval, say, $[6, 11]$, or over two intervals $[7, 8]$ and $[15, 19]$, or even three intervals $[200, 201]$, $[300, 301]$, $[400, 403]$.

   (a) (5 points) Give a set of jobs that is not feasible, i.e., no schedule can finish all jobs on time.

   (b) (20 points) Design an efficient algorithm to determine whether a job set J is feasible or not. Let D be the maximum deadline among all jobs, i.e., $D = \max_{i=1}^{n} d_i$, you should analyze the running time in terms of n and D. (Tips, you need to prove that if the input job set J is feasible, then the algorithm can find a feasible schedule for J.)

   (c) (Bonus: 5 points) Design an efficient algorithm with the running time only in terms of n.

4. (25 points) Makespan Minimization Given m identical machines and n jobs with size $p_1, p_2, ..., p_n$. How to find a feasible schedule of these n jobs on m machines to minimize the *makespan*: the maximized completion time among all m machines?

   Recall that we have introduced two greedy approaches in the lecture.

   - GREEDY: Schedule jobs in an arbitrary order, and we always schedule jobs to the earliest finished machine.
   - LPT: Schedule jobs in the decreasing order of their size, and we always schedule jobs to the earliest finished machine.

   We have proved that GREEDY is 2-approximate and LPT is 1.5-approximate, can we finish the following tasks? (Please write down complete proofs.)

   (a) (10 points) Complete the proof that LPT is 4/3-approximate, (i.e., $\mathsf{LPT} \leq 4/3 \cdot \mathsf{OPT}$).

   (b) (10 points) Prove that GREEDY is $(2 - 1/m)$-approximate, (i.e., $\mathsf{GREEDY} \leq (2 - 1/m) \cdot \mathsf{OPT}$).

   (c) (5 points) Prove that LPT is $(4/3 - 1/3m)$-approximate, (i.e., $\mathsf{LPT} \leq (4/3 - 1/3m) \cdot \mathsf{OPT}$).

5. How long does it take you to finish the assignment (include thinking and discussing)? Give a score (1,2,3,4,5) to the difficulty. Do you have any collaborators? Write down their names here.