Algorithm Design and Analysis
Assignment 7

1. Show that the following problem is NP-Complete.

   MAXIMUM COMMON SUBGRAPH Input: Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$; a budget b. Output: Decide whether there exist two set of nodes $V_1' \subseteq V_1$ and $V_2' \subseteq V_2$ whose deletion leaves at least b nodes in each graph, and makes the two graphs identical.

   Solution. Given G_1, G_2, V_1', and V_2', we can efficiently delete these vertices from G_1 and G_2 and check the two remaining subgraph are identical and have at least b nodes in $O(n^2)$ time. (Notice that identical is different from isomorphic.) It proves that MAXIMUM COMMON SUBGRAPH is in NP.

   Consider the NP-Complete problem k-CLIQUE, which ask us to find a size k clique in a given graph G. Let us consider the MAXIMUM COMMON SUBGRAPH where $G_1 = G$, $b = k$, and G_2 be the size k clique. For any yes instance of the k-CLIQUE, we can find a size k clique, which is a size k subgraph of G_1 and is identical to G_2. Therefore, the input we construct before is also a yes instance of MAXIMUM COMMON SUBGRAPH. On the other hand, if the input of MAXIMUM COMMON SUBGRAPH is yes, then because the subgraph contains at least b nodes, it must be G_2, and G_2 must be a subgraph of G_1, which is a size k clique. Therefore, G should be a yes instance of k-CLIQUE. Thus far, we give a karp reduction from k-CLIQUE to MAXIMUM COMMON SUBGRAPH, and prove MAXIMUM COMMON SUBGRAPH is NP-Complete. □

2. SINGLE EXECUTION TIME SCHEDULING (SS). Given a set S of n jobs, a relation $\prec$ on S, a number of processors k and a time limit t. Does there exist a function f from S to $\{0, 1, ..., t-1\}$ such that

   1. $f^{-1}(i)$ has at most k members, and
   2. if $J \prec J'$, then $f(J) < f(J')$?

   For a verbal description of the problem, we need to assign those n jobs to those k processors. Each processor takes one unit of time to complete each job. $f(J)$ is the starting time for job J, and job J ends at time $f(J)+1$. The first requirement above says that at most k jobs can be executed simultaneously. For two jobs J and J', $J \prec J'$ indicates J must be finished before executing J'. This is captured by the second requirement. We are deciding if we can schedule the n jobs on those k processors such that all of them are finished before time t.

   Now consider a more complex version: Single execution time scheduling with variable number of processors (SSV). Given a set S of n jobs, a relation $\prec$ on S, a time limit t, and a sequence of integers $c_0, c_1 ..., c_{t-1}$, where $\sum_{i=0}^{t-1} c_i = n$, does there exist a function f from S to $0, 1, ..., t-1$ such that

   1. $f^{-1}(i)$ has exactly c_i members, and
   2. if $J \prec J'$, then $f(J) < f(J')$?

   Show that (SS) is NP-Complete. You may follows the three steps below:

   (a) Show (SS) is in NP.

   (b) Show that there is a Karp reduction from SSV to SS: SSV $\leq_k$ SS.

   (c) Show that there is a Karp reduction from 3SAT to SSV: 3SAT $\leq_k$ SSV.

   Hint for part (c):

   - For each variable x_i ($i = 1, \ldots, n$) in the 3SAT instance, create two gadgets corresponding to the two Boolean assignments to the variable as follows:
     - Gadget for $x_i =$ **true**: create $n+2$ jobs $x_{i0}, x_{i1}, \ldots, x_{in}, y_i$ with $x_{i0} \prec x_{i1} \prec \cdots \prec x_{in}$ and $x_{i(i-1)} \prec y_i$;
     - Gadget for $x_i =$ **false**: create $n+2$ jobs $\bar{x}_{i0}, \bar{x}_{i1}, \ldots, \bar{x}_{in}, \bar{y}_i$ with $\bar{x}_{i0} \prec \bar{x}_{i1} \prec \cdots \prec \bar{x}_{in}$ and $\bar{x}_{i(i-1)} \prec \bar{y}_i$.
     - Intuitive idea: we may imagine x_i (or $\bar{x}_i$) to be true if and only if x_{i0} (or $\bar{x}_{i0}$, respectively) is executed at time 0, all the other jobs are used to control the requirement of the 3SAT problem.

- For each clause C_j $(j = 1, \ldots, m)$, create 7 jobs $c^j_{ttt}, c^j_{ttf}, c^j_{tft}, c^j_{ftt}, c^j_{tff}, c^j_{ftf}, c^j_{fft}$ corresponding to the seven possibilities of the values for the three literals in the clause (that makes the clause evaluated to **true**). For example, c^j_{ttf} corresponds to that the first and the second literals are **true** and the third literal is **false**. Then, link each of the seven jobs to the last job in the variable gadget accordingly. For example, if we have a clause $C_j = x_3 \vee \bar{x}_4 \vee x_7$ and consider the job c^j_{ttf} (which corresponds to $x_3 =$ **true**, $\bar{x}_4 =$ **true** and $x_7 =$ **false**), we have $x_{3n} \prec c^j_{ttf}$, $\bar{x}_{4n} \prec c^j_{ttf}$ and $\bar{x}_{7n} \prec c^j_{ttf}$.
- Consider $t = n + 2$. Setup the values for $c_0, c_1, \ldots, c_{t-1}$ appropriately such that, in order to finish all the tasks, we must have the followings:
  - For each $i = 1, \ldots, n$, we must have either $f(x_{i0}) = 0, f(\bar{x}_{i0}) = 1$ or $f(x_{i0}) = 1, f(\bar{x}_{i0}) = 0$. The former case will represent $x_i =$ **true** and the latter case will represent $x_i =$ **false**;
  - For the 7 jobs corresponding to each clause, exactly one job must be executed at time n, and the remaining 6 jobs must be executed at time $n + 1$. The job executed at time n will represent the values for the three literals in the clause in a satisfying Boolean assignment.

Solution. Please refer to the paper of J.D. ULLMAN. (P2) is our problem (SS) in the paper, they prove (P2) is NP-Complete via (P4) (i.e. (SSV)). □

3. 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.