

1. Tenemos un listado de proyectos a realizar, que tienen tiempo de inicio y tiempo de finalización. Contamos además con K equipos que pueden realizar los proyectos. Cada proyecto puede estar asignado a un único equipo (o ninguno, si no se realiza). No hay distinción entre los equipos (todos pueden realizar todos los proyectos). Un mismo equipo no puede estar trabajando en dos proyectos a la vez, aunque varios equipos pueden estar trabajando en proyectos diferentes en simultáneo sin problemas. Implementar un algoritmo **greedy** que determine la asignación de proyectos a los diferentes equipos, de tal manera que se maximice la cantidad de proyectos a realizar. Indicar y justificar la complejidad del algoritmo implementado. Indicar por qué se trata, en efecto, de un algoritmo greedy. El algoritmo, ¿es óptimo? si lo es, justificar brevemente, sino dar un contraejemplo (es muy probable que el algoritmo NO sea óptimo, así que pensar bien por un contraejemplo antes de decir que lo es).

2. Implementar un algoritmo que, por backtracking, resuelva el problema del ejercicio 1.

3. Queremos determinar si un texto ininteligible, sin espacios, es en realidad un texto de un idioma, al que se le borraron los espacios (por ejemplo, "holaquehaces"). Contamos con un conjunto de las k palabras (muy cortas) del idioma en cuestión.

Implementar un algoritmo que, utilizando programación dinámica, reciba el texto, y el conjunto de palabras del idioma, y determine si, en efecto, se trata de un texto del idioma, sin los espacios. Por ejemplo, para "argentinacampeon" debe devolver **true**, mientras que para "awanteboke" debe devolver **false**. La resolución de este ejercicio puede ser similar a la resolución del TP2 (curso regular)/Parte II (cursos anual y asincrónico). El ejercicio puede resolverse incluso de una forma mejor (que a algunos puede resultarles más simple). Indicar y justificar la complejidad del algoritmo implementado.

4. Grace, docente de un primer grado de un colegio primario, tiene como tarea armar la fila del acto del 9 de julio. Como no quiere que los chicos se distraigan, quiere que cada alumno quede entre dos personas de las cuales no es amigo. *El problema de la fila del primario* (en versión de problema de decisión) puede formularse como: “Dado un **listado de alumnos**, y un **diccionario** que indica qué alumnos son amigos de cuál, ¿existe una forma de poner en fila a los alumnos para que no haya ningún par de amigos consecutivos?”

Demostrar que *El problema de la fila del primario* es un problema NP-Completo. Para esto, recomendamos recordar que el problema de *Camino Hamiltoniano* es un problema NP-Completo. También, recomendamos recordar cómo fue que demostramos en clase que *K-Clique* es un problema NP-Completo (con la ayuda de *Independent Set*).

5. Realizar el seguimiento de aplicar el algoritmo de Ford-Fulkerson a la red del dorso, explicando (en caso de ser necesario) cuáles son las modificaciones previas a realizarle. Luego de realizar dicho seguimiento, suponer que se agrega una arista del vértice v_2 al vértice v_3 , de capacidad 9. Trabajando sobre la red residual resultante del seguimiento previo, continuar el seguimiento con este agregado. Dar el resultado final del flujo, considerando la red original (incluyendo el agregado de la arista mencionada, pero sin las modificaciones que hayas hecho inicialmente).

1. Tenemos un listado de proyectos a realizar, que tienen tiempo de inicio y tiempo de finalización. Contamos además con K equipos que pueden realizar los proyectos. Cada proyecto puede estar asignado a un único equipo (o ninguno, si no se realiza). No hay distinción entre los equipos (todos pueden realizar todos los proyectos). Un mismo equipo no puede estar trabajando en dos proyectos a la vez, aunque varios equipos pueden estar trabajando en proyectos diferentes en simultáneo sin problemas. Implementar un algoritmo **greedy** que determine la asignación de proyectos a los diferentes equipos, de tal manera que se maximice la cantidad de proyectos a realizar. Indicar y justificar la complejidad del algoritmo implementado. Indicar por qué se trata, en efecto, de un algoritmo greedy. El algoritmo, ¿es óptimo? si lo es, justificar brevemente, sino dar un contraejemplo (es muy probable que el algoritmo NO sea óptimo, así que pensar bien por un contraejemplo antes de decir que lo es).

2. Implementar un algoritmo que, por backtracking, resuelva el problema del ejercicio 1.

3. Queremos determinar si un texto ininteligible, sin espacios, es en realidad un texto de un idioma, al que se le borraron los espacios (por ejemplo, "holaquehaces"). Contamos con un conjunto de las k palabras (muy cortas) del idioma en cuestión.

Implementar un algoritmo que, utilizando programación dinámica, reciba el texto, y el conjunto de palabras del idioma, y determine si, en efecto, se trata de un texto del idioma, sin los espacios. Por ejemplo, para "argentinacampeon" debe devolver **true**, mientras que para "awanteboke" debe devolver **false**. La resolución de este ejercicio puede ser similar a la resolución del TP2 (curso regular)/Parte II (cursos anual y asincrónico). El ejercicio puede resolverse incluso de una forma mejor (que a algunos puede resultarles más simple). Indicar y justificar la complejidad del algoritmo implementado.

4. Grace, docente de un primer grado de un colegio primario, tiene como tarea armar la fila del acto del 9 de julio. Como no quiere que los chicos se distraigan, quiere que cada alumno quede entre dos personas de las cuales no es amigo. *El problema de la fila del primario* (en versión de problema de decisión) puede formularse como: “Dado un **listado de alumnos**, y un **diccionario** que indica qué alumnos son amigos de cuál, ¿existe una forma de poner en fila a los alumnos para que no haya ningún par de amigos consecutivos?”

Demostrar que *El problema de la fila del primario* es un problema NP-Completo. Para esto, recomendamos recordar que el problema de *Camino Hamiltoniano* es un problema NP-Completo. También, recomendamos recordar cómo fue que demostramos en clase que *K-Clique* es un problema NP-Completo (con la ayuda de *Independent Set*).

5. Realizar el seguimiento de aplicar el algoritmo de Ford-Fulkerson a la red del dorso, explicando (en caso de ser necesario) cuáles son las modificaciones previas a realizarle. Luego de realizar dicho seguimiento, suponer que se agrega una arista del vértice v_2 al vértice v_3 , de capacidad 9. Trabajando sobre la red residual resultante del seguimiento previo, continuar el seguimiento con este agregado. Dar el resultado final del flujo, considerando la red original (incluyendo el agregado de la arista mencionada, pero sin las modificaciones que hayas hecho inicialmente).

