Proyecto de curso

El siguiente texto plantea el proyecto del curso de *"Teoría de la Computación"* a dictarse en el segundo semestre de 2018 por Lucía Tejería y Leonardo Val. La cátedra se reserva el derecho de corregir, modificar y ampliar el siguiente documento según crean conveniente.

Longest common subsequence

El problema de la *máxima subsecuencia común* (o LCS) es un problema clásico de computación. Dadas dos listas, se desea encontrar la subsecuencia más larga que ambas tengan en común. Ésta subsecuencia no es una sublista (o *substring*), dado que sus elementos no tienen por qué aparecer contiguos en ambas listas. Sin embargo sus elementos sí deben aparecer en el mismo orden en ambas listas dadas.

El problema tiene al menos dos algoritmos que pueden solucionarlo. El primero sería el algoritmo por fuerza bruta o *ingenuo*, que plantea todas las posibles subsequencias de la lista más corta, y toma la más larga que verifique ser una subsecuencia válida de la otra lista. El segundo utiliza *programación dinámica* para lograr un orden polinomial.

Planteo del trabajo

El proyecto del curso 2018 de *Teoría de la Computación* se trata de analizar teórica y empíricamente al menos dos algoritmos para resolver el problema LCS. Cada equipo deberá:

- Estudio teórico del tiempo de ejecución:
  - Obtener o escribir el pseudocódigo para un algoritmo ingenuo y un algoritmo de programación dinámica que resuelvan el LCS.
  - Estudiar dichos algoritmos para caracterizar lo más ajustadamente posible su función de tiempo y definir su orden de ejecución.
- Estudio empírico del tiempo de ejecución:
  - Implementar los algoritmos en Python 3.
  - Programar la generación aleatoria de casos de prueba para ambos algoritmos anteriores.
  - Definir dos hipótesis alternativas para las funciones de tiempo de cada uno de los algoritmos implementados.
  - Instrumentar los algoritmos para medir la cantidad de pasos ejecutados. En el caso del algoritmo de programación dinámica, interesa la cantidad de accesos a la matriz utilizada.
  - Ejecutar los algoritmos de manera controlada con diferentes casos de prueba aleatorios, midiendo la cantidad de pasos con las versiones instrumentadas y el tiempo real con las versiones sin instrumentar.
  - Ajustar los parámetros de las hipótesis de función de tiempo mediante regresión lineal. Para esto se puede utilizar la biblioteca `numpy`.
  - Para cada función de tiempo ajustada, calcular el tiempo teórico en cada caso y su diferencia con los valores reales (llámese *error*).
  - Verificar si la distribución de los errores para cada función de tiempo ajustada es binomial. Para esto se puede utilizar la biblioteca `scipy`.
- Documentar todo el desarrollo del proyecto.
  - Graficar los resultados teóricos y empíricos.
  - Discutir y analizar los resultados obtenidos.

La entrega se realizará vía Webasignatura el 23 de Noviembre a las 18:00 horas. Cada equipo deberá subir un archivo comprimido con la documentación en PDF, el código fuente Python 3 utilizado (sin incluir binarios compilados) y los datos de las ejecuciones en CSV.

Referencias

- [BERGROTH2000] L. Bergroth, H. Hakonen & T. Raita; *"A survey of longest common subsequence algorithms"*, Proceedings Seventh International Symposium on String Processing and Information Retrieval. SPIRE 2000, Setiembre 2000. DOI:10.1109/SPIRE.2000.878178. ISBN 0-7695-0746-8.
- [NUMPY2018] The SciPy community; `numpy.linalg.lstsq`, NumPy v1.15 Manual, Noviembre 2018.
- [SCIPY2018] The SciPy community; `scipy.stats.normaltest`, SciPy v1.1.0 Reference Guide, Mayo 2018.

Fin