# Trabajo Práctico 2: Que parezca programación dinámica

El presente trabajo busca evaluar el desarrollo y análisis de un algoritmo de Programación Dinámica. La fecha de entrega del mismo es el 5/05.

## Introducción

Luego de haber ayudado al Gringo y a Amarilla Pérez a encontrar a *la rata* que les estaba robando dinero ganado a fuerza de sudor y sangre (no de ellos), hemos sido ascendidos.

Amarilla detectó que hay un soplón en la organización, que parece que se contacta con alguien de la policía. En general eso no sería un problema, ya que la mafia trabaja en un distrito donde media policía está arreglada. El problema es que no parecería ser el caso. El soplón parece estarse contactando con mensajes encriptados.

La organización no está conformada por novatos; no es la primera vez que algo así sucede. Ya han desencriptado mensajes de este estilo, y han *charlado amablemente* con su emisor. El problema es que en este caso parece ser más complicado. El soplón de esta oportunidad parece encriptar todas las palabras juntas, sin espacios. Eso complica más la desencriptación y validar que el mensaje tenga sentido.

No interesa saber quién es el soplón (de momento), sino más bien qué información se está filtrando. El área de desencriptación se encargará de intentar dar posibles resultados, y nosotros debemos validar si, en principio, es un posible mensaje. Es decir, si nos dan una *cadena desencriptada* que diga `"estanocheenelmuellealassiete"`, este sería un posible mensaje, el cual correspondería a `"esta noche en el muelle a las siete"`, mientras que `"estamikheestado"` no lo es, ya que no podemos separar de forma de generar todas palabras del idioma español (en cambio, si fuera `"estamiheestado"` podría ser `"esta mi he estado`). No es nuestra labor analizar si el texto tiene potencialmente sentido o no, de eso se encargará otro área.

El Gringo nos sugirió que usemos **programación dinámica** para esto. Y, en general, cuando lo sugiere, es porque es necesario. Eso, y que no seguir con su sugerencia puede implicar desayunar con los peces.

A considerar: Dado que estamos trabajando en español, no es necesario considerar el caso que pueda haber más de una opción para la separación (por ejemplo, esto podría pasar con `"estamosquea"`, que puede separarse en `"estamos que a"` o bien `"esta mosquea"`), ya que son pocos casos, muy forzados y poco probables. Si el trabajo fuera en un idioma germánico (como el alemán) esto podría ser un poco más problemático.

## Consigna

1. Hacer un análisis del problema, plantear la ecuación de recurrencia correspondiente y proponer un algoritmo por programación dinámica que obtenga la solución óptima al problema planteado: Dado el listado $n$ de palabras, y una *cadena desencriptada*, determinar si es un *posible mensaje* (es decir, se lo puede separar en palabras del idioma), o no. Si es posible, determinar cómo sería el mensaje.
2. Demostrar que la ecuación de recurrencia planteada en el punto anterior en efecto nos asegura encontrar una solución, si es que la hay (y si no la hay, lo detecta).
3. Escribir el algoritmo planteado. Describir y justificar la complejidad de dicho algoritmo. Considerar analizar por separado cada uno de los algoritmos que se implementen (programación dinámica y reconstrucción), y luego llegar a una conclusión final.
4. Analizar si (y cómo) afecta a los tiempos del algoritmo planteado la variabilidad de los valores (mensajes, palabras del idioma, largos de las palabras, etc.).
5. Realizar ejemplos de ejecución para encontrar soluciones y corroborar lo encontrado. Adicionalmente, el curso proveerá con algunos casos particulares para que puedan usar inicialmente para validar.
6. Hacer mediciones de tiempos para corroborar la complejidad teórica indicada. Agregar los casos de prueba necesarios para dicha corroboración (generando sus propios sets de datos). Esta corroboración empírica debe realizarse confeccionando gráficos correspondientes, y utilizando la técnica de cuadrados mínimos. Para esto, proveemos una explicación detallada, en conjunto de ejemplos.
7. Agregar cualquier conclusión que les parezca relevante.

## Entrega

Debe enviarse al corrector asignado, por mail o slack, el link al repositorio donde se encuentre el código fuente, y donde debe encontrarse el informe en formato PDF, que debe seguir los lineamientos establecidos en el TP1. En este caso, la forma de ejecutar el programa debe ser:

```
./tp2 ruta/a/listado-palabras.txt
```

O bien:

```
python3 tp2.py ruta/a/listado-palabras.txt
```

En cualquiera de los 2 casos, se debe leer por entrada estándar las posibles cadenas que podrían ser mensajes a detectar (una por línea). Es decir:

```
python3 tp2.py ruta/a/listado.txt < posibles_mensajes.txt
```

Nuevamente, la nota del trabajo práctico tendrá en cuenta tanto la presentación y calidad de lo presentado, como también el desarrollo del trabajo. No será lo mismo un trabajo realizado con lo mínimo indispensable, que uno bien presentado, analizado, y probado con diferentes volúmenes, set de datos, o estrategias de generación de sets, en el caso que corresponda.