

¡Bienvenidos al labo de Algo III!

Laboratorio de Algoritmos y Estructuras de Datos III

2^{do} Cuatrimestre de 2018

Los docentes del laboratorio somos:

- Alejandro Candiotti,
- Federico Pousa,
- Gonzalo Lera Romero y
- Matias Millasson.

Los días de laboratorio van a ser los viernes de 17:30 a 21:30 hs, y **trataremos** de seguir el siguiente esquema de clases:

- de 17:30 a 18:00 hs: consultas,
- de 18:00 a 20:30 hs: clase y/o presentación de TPs y
- de 20:30 a 21:30 hs: consultas y/o resolución de ejercicios.

Objetivo del Labo

Qué NO es el labo:

- El Labo no es un repaso de la teórica y la práctica con la excusa para evaluar una vez más.
- El Labo no es una materia aparte.

Objetivo del Labo

Qué NO es el labo:

- El Labo no es un repaso de la teórica y la práctica con la excusa para evaluar una vez más.
- El Labo no es una materia aparte.

El Labo es una parte más de Algo 3 con algunos objetivos particulares:

- Implementar los conceptos particulares y meter mano en cuestiones prácticas.
- Enmarcar los temas de Algo 3 en contextos de aplicaciones en el mundo real.
- Reforzar cuestiones metodológicas referidas al desarrollo de un trabajo de investigación y a su comunicación.

Los TPs

Para aprobar el laboratorio hay que aprobar 3 trabajos prácticos.
En general (a menos que se especifique lo contrario):

- El primer TP es individual, el resto se realiza de a grupos.

Los TPs

Para aprobar el laboratorio hay que aprobar 3 trabajos prácticos.
En general (a menos que se especifique lo contrario):

- El primer TP es individual, el resto se realiza de a grupos.
- Cada grupo debe tener cuatro alumnos.

Los TPs

Para aprobar el laboratorio hay que aprobar 3 trabajos prácticos.
En general (a menos que se especifique lo contrario):

- El primer TP es individual, el resto se realiza de a grupos.
- Cada grupo debe tener cuatro alumnos.
- Los TPs tienen 3 notas. Desaprobado, Aprobado y Promocionado.

Los TPs

Para aprobar el laboratorio hay que aprobar **3 trabajos prácticos**.
En general (a menos que se especifique lo contrario):

- El primer TP es individual, el resto se realiza de a grupos.
- Cada grupo debe tener cuatro alumnos.
- Los TPs tienen 3 notas. Desaprobado, Aprobado y Promocionado.
- Para promocionar el laboratorio se deben tener 2 TPs promocionados y uno de ellos debe ser el tercero.

Los TPs

Para aprobar el laboratorio hay que aprobar **3 trabajos prácticos**.
En general (a menos que se especifique lo contrario):

- El primer TP es individual, el resto se realiza de a grupos.
- Cada grupo debe tener cuatro alumnos.
- Los TPs tienen 3 notas. Desaprobado, Aprobado y Promocionado.
- Para promocionar el laboratorio se deben tener 2 TPs promocionados y uno de ellos debe ser el tercero.
- Hay **una fecha de entrega** y **una fecha de reentrega**.

Los TPs

Para aprobar el laboratorio hay que aprobar **3 trabajos prácticos**.
En general (a menos que se especifique lo contrario):

- El primer TP es individual, el resto se realiza de a grupos.
- Cada grupo debe tener cuatro alumnos.
- Los TPs tienen 3 notas. Desaprobado, Aprobado y Promocionado.
- Para promocionar el laboratorio se deben tener 2 TPs promocionados y uno de ellos debe ser el tercero.
- Hay una fecha de entrega y una fecha de reentrega.
- Los horarios serán binariamente e irreversiblemente estrictos.

Los TPs

Para aprobar el laboratorio hay que aprobar **3 trabajos prácticos**.
En general (a menos que se especifique lo contrario):

- El primer TP es individual, el resto se realiza de a grupos.
- Cada grupo debe tener cuatro alumnos.
- Los TPs tienen 3 notas. Desaprobado, Aprobado y Promocionado.
- Para promocionar el laboratorio se deben tener 2 TPs promocionados y uno de ellos debe ser el tercero.
- Hay una fecha de entrega y una fecha de reentrega.
- Los horarios serán binariamente e irreversiblemente estrictos.
- Toda reentrega deberá estar acompañada por un **informe de modificaciones** detallando brevemente las diferencias entre las dos entregas (cambios, agregados y/o partes eliminadas).

Los TPs

Para aprobar el laboratorio hay que aprobar **3 trabajos prácticos**.
En general (a menos que se especifique lo contrario):

- El primer TP es individual, el resto se realiza de a grupos.
- Cada grupo debe tener cuatro alumnos.
- Los TPs tienen 3 notas. Desaprobado, Aprobado y Promocionado.
- Para promocionar el laboratorio se deben tener 2 TPs promocionados y uno de ellos debe ser el tercero.
- Hay una fecha de entrega y una fecha de reentrega.
- Los horarios serán binariamente e irreversiblemente estrictos.
- Toda reentrega deberá estar acompañada por un *informe de modificaciones* detallando brevemente las diferencias entre las dos entregas (cambios, agregados y/o partes eliminadas).
- Reentregar un TP significa corregir **todo** el TP.

Los TPs

Para aprobar el laboratorio hay que aprobar **3 trabajos prácticos**.
En general (a menos que se especifique lo contrario):

- El primer TP es individual, el resto se realiza de a grupos.
- Cada grupo debe tener cuatro alumnos.
- Los TPs tienen 3 notas. Desaprobado, Aprobado y Promocionado.
- Para promocionar el laboratorio se deben tener 2 TPs promocionados y uno de ellos debe ser el tercero.
- Hay una fecha de entrega y una fecha de reentrega.
- Los horarios serán binariamente e irreversiblemente estrictos.
- Toda reentrega deberá estar acompañada por un *informe de modificaciones* detallando brevemente las diferencias entre las dos entregas (cambios, agregados y/o partes eliminadas).
- Reentregar un TP significa corregir **todo** el TP.
- **Quienes tengan aprobados los TPs del cuatrimestre pasado, pueden elegir no cursarlos este cuatrimestre.(pero lo tienen que avisar!)**

Si bien toda la materia se cursa y se aprueba en español, es muy recomendable el manejo de un inglés técnico básico.

- Cursos de inglés en la facu.
- Becas o descuentos por ser alumnos de la UBA (en el CUI por ejemplo).
- Alguna otra opción más artesanal como Duolingo o armar un grupo de estudio que sigan la bibliografía en inglés.

Lenguajes de programación

- El **lenguaje de programación** del TP1 es `c++`
- Esto solo aplica a los códigos específicos de los algoritmos necesarios para los tps.
- Toda la experimentación, la generación de gráficos, etc. es posible (y recomendable!) hacerlo con otros lenguajes.
- No hay que reinventar la rueda, pero sí hay que hacer sus propios algoritmos particulares.
- Se puede usar todo lo que esté disponible en la STL y tomar *[http : //www.cplusplus.com/reference/stl/](http://www.cplusplus.com/reference/stl/)* como referencia para el uso y complejidad.

Lenguajes de programación

- El **lenguaje de programación** del TP1 es $c++$
- Esto solo aplica a los códigos específicos de los algoritmos necesarios para los tps.
- Toda la experimentación, la generación de gráficos, etc. es posible (y recomendable!) hacerlo con otros lenguajes.
- No hay que reinventar la rueda, pero sí hay que hacer sus propios algoritmos particulares.
- Se puede usar todo lo que esté disponible en la STL y tomar *[http : //www.cplusplus.com/reference/stl/](http://www.cplusplus.com/reference/stl/)* como referencia para el uso y complejidad.

El **procesador de texto** a utilizar también es a elección del grupo, aunque (sinceramente y por su bien) se sugiere el uso de \LaTeX .

Una *clásica* comunicación científica sobre una resolución experimental de un problema en Ciencias de la Computación puede tener el siguiente formato:

Una *clásica* comunicación científica sobre una resolución experimental de un problema en Ciencias de la Computación puede tener el siguiente formato:

- Introducción: Se introduce el problema, se muestra un review de la literatura y otros aspectos introductorios como posibles aplicaciones.

Una *clásica* comunicación científica sobre una resolución experimental de un problema en Ciencias de la Computación puede tener el siguiente formato:

- Introducción: Se introduce el problema, se muestra un review de la literatura y otros aspectos introductorios como posibles aplicaciones.
- Desarrollo: Posiblemente dividido en subsecciones donde se muestran los modelos considerados, los algoritmos propuestos y los resultados teóricos obtenidos, entre otras cosas.

Una *clásica* comunicación científica sobre una resolución experimental de un problema en Ciencias de la Computación puede tener el siguiente formato:

- Introducción: Se introduce el problema, se muestra un review de la literatura y otros aspectos introductorios como posibles aplicaciones.
- Desarrollo: Posiblemente dividido en subsecciones donde se muestran los modelos considerados, los algoritmos propuestos y los resultados teóricos obtenidos, entre otras cosas.
- Experimentación: Se definen instancias de fundamentado interés y se experimenta con las propuestas mencionadas en el desarrollo.

Una *clásica* comunicación científica sobre una resolución experimental de un problema en Ciencias de la Computación puede tener el siguiente formato:

- Introducción: Se introduce el problema, se muestra un review de la literatura y otros aspectos introductorios como posibles aplicaciones.
- Desarrollo: Posiblemente dividido en subsecciones donde se muestran los modelos considerados, los algoritmos propuestos y los resultados teóricos obtenidos, entre otras cosas.
- Experimentación: Se definen instancias de fundamentado interés y se experimenta con las propuestas mencionadas en el desarrollo.
- Discusión y conclusiones: Se discuten los resultados observados en la sección anterior y se da un cierre general.

¿Qué se espera de los TPs?

Para cada ejercicio de los TPs se esperan los siguientes puntos:

- 1.** Describir detalladamente el problema a resolver dando ejemplos del mismo y sus soluciones.

¿Qué se espera de los TPs?

Para cada ejercicio de los TPs se esperan los siguientes puntos:

1. Describir detalladamente el problema a resolver dando ejemplos del mismo y sus soluciones.

2. Explicar de forma clara, sencilla, estructurada y concisa, las ideas desarrolladas para la resolución del problema. Para esto se pide utilizar pseudocódigo y lenguaje coloquial combinando adecuadamente ambas herramientas (**¡sin usar código fuente!**). Se debe también justificar por qué el procedimiento desarrollado resuelve efectivamente el problema.

¿Qué se espera de los TPs?

3. Deducir una **cota de complejidad temporal** del algoritmo propuesto, en función de los parámetros que se consideren correctos y **justificar** por qué el algoritmo desarrollado para la resolución del problema cumple la cota dada.

¿Qué se espera de los TPs?

3. Deducir una **cota de complejidad temporal** del algoritmo propuesto, en función de los parámetros que se consideren correctos y **justificar** por qué el algoritmo desarrollado para la resolución del problema cumple la cota dada.

4. Dar un código fuente claro que implemente la solución propuesta. El mismo no sólo debe ser correcto sino que además debe seguir las **buenas prácticas de la programación** (comentarios pertinentes, nombres de variables apropiados, estilo de indentación coherente, modularización adecuada, etc.).

¿Qué se espera de los TPs?

5. Realizar una **experimentación computacional** para medir la performance del programa implementado. Para ello se debe **preparar un conjunto de casos de test** que permitan observar los tiempos de ejecución en función de los parámetros de entrada. Deberán desarrollarse tanto experimentos con instancias **aleatorias (detallando cómo fueron generadas)** como experimentos con instancias **particulares** (de peor/mejor caso en tiempo de ejecución, por ejemplo). Presentar **adecuadamente en forma gráfica** una comparación entre los tiempos medidos y la complejidad teórica calculada y extraer conclusiones de la experimentación.

Hagamos la prueba con el *Partition Problem*

- Una fuerza bruta *inteligente*.
- Se van armando soluciones parciales para el problema explorando todas las opciones.
- Cuando una solución parcial no puede progresar con seguridad, se elimina esa opción y se vuelve sobre los pasos anteriores (backtrack).
- La exploración de todas las decisiones da lugar a un *árbol de backtracking* que se puede *podar* para ayudar a la búsqueda.

Más allá que siga siendo una enumeración de soluciones, hay varios aspectos que influyen en la performance:

- Podas por factibilidad.
- Diseño de podas por optimalidad (para problemas de optimización).
- Otras decisiones particulares como reordenamiento de las decisiones.

Las 8 reinas

Tenemos que poner 8 reinas en un tablero de ajedrez sin que se amenacen mutuamente.

Aprovechemos la técnica de Backtracking para mejorar las soluciones por fuerza bruta.

Enunciado

Cronograma sugerido para el tp:

- 24 de Agosto: Presentación TP1.
- 31 de Agosto: Desarrollo del problema y de los diferentes algoritmos (sin dinámica). Código de fuerza bruta.
- 7 de Septiembre: Código backtracking y dinámica. Definición de instancias y primeras experimentaciones.
- 14 de Septiembre: Experimentación en conjunto. Discusión final.
- 16 de Septiembre: Entrega de TP1.

¡Eso es todo por hoy!