

Tarea 2 - Cerradura Convexa

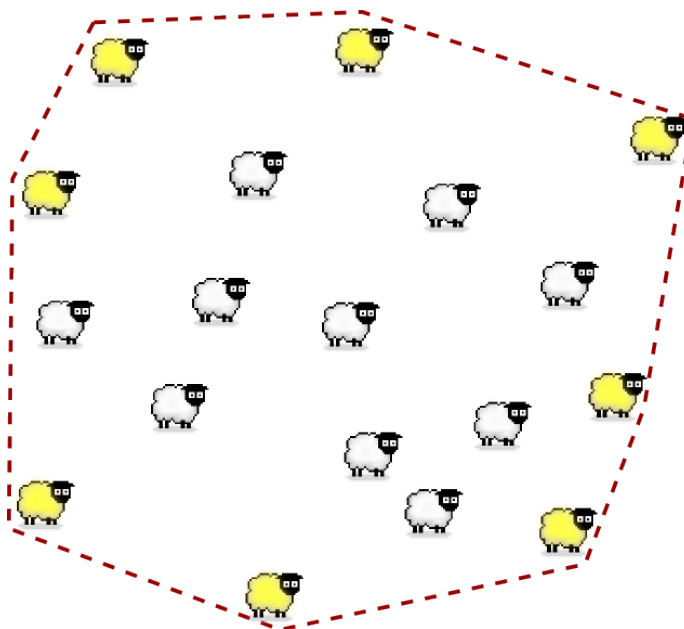
Profesora: Nancy Hitschfeld K.

Ayudante: Michel Llorens

1. Introducción

La cerradura convexa, aparte de ser un conjunto matemático de puntos que cumplen ciertas condiciones, es algo que va más allá de la teoría y se utiliza activamente en distintas aplicaciones. Tales como reconocimiento de patrones, procesamiento de imágenes, información geográfica e incluso cosas como la teoría de juegos.

Por todo ello y mucho más, hoy en día existen al menos 8 formas distintas de implementar dicho algoritmo y el objetivo principal de esta tarea, será probar al menos dos de ellas utilizando como base su Tarea 1.



2. Elementos a implementar.

Esta tarea será implementada sobre su Tarea 1 y, por tanto, **toda esta tarea debe estar implementada en C++**.

La tarea está dividida en tres secciones:

1. Debe usted implementar el algoritmo de **Gift Wrapping**, el cual debe recibir como argumento un listado de puntos y debe retornar el polígono que representa la cerradura. Puede usted guiarse por la siguiente firma: `Polygon giftWrapping(Point[] cloud)`.
2. Como Gift Wrapping tiene peor caso $O(n^2)$ y podemos estar trabajando con nubes de millones de puntos, es necesario poder utilizar otro algoritmo de cerradura convexa que tenga un mejor peor caso. Por ello, usted debe implementar **UNO** de los siguientes algoritmos (que siga la misma firma que Gift Wrapping):
 - Graham scan
 - Quickhull
 - Divide and conquer
 - Monotone chain
 - Incremental convex hull algorithm
 - The ultimate planar convex hull algorithm
 - Chan's algorithm
3. Una vez implementado ambos métodos, es necesario compararlos para ver cuál es más eficiente con respecto a la cantidad de puntos. Por ello, debe implementar un método que genere una lista aleatoria de puntos de tamaño 2^{10} y a continuación otro método debe utilizar dicha lista para calcular la cerradura convexa con ambos algoritmos (y en consecuencia, medir sus tiempos). **OJO: También debe comparar que una vez terminado el procesamiento de ambas cerraduras, estas deben ser idénticas, ya que la cerradura convexa es única.**

5. Información extra

- Los resultados de sus comparaciones además de estar en un main (u otro archivo con fin de realizar el testing) deben venir en un archivo de texto llamado **ANALISIS**, donde debe indicar si tomó algún supuesto en las implementaciones y bajo qué condiciones obtuvo sus resultados, además de estos últimos.
- Si usted lo desea, puede realizar pruebas con distintos tamaños de listas a fin de poder generar un gráfico, el cual se puede adjuntar a la tarea.
- La fecha de entrega así como los contenidos de la tarea podrían llegar a modificarse según las necesidades del curso y cómo este avance, pero de ocurrir esto se notificará en clases y posteriormente se verá reflejado en U-Cursos.
- Debe incluir **README** sobre cómo ejecutar su programa.