

Tarea 1

Profesor: Diego Arroyuelo
Ayudantes: Tomás Berrios, Alejandro Vilches
tomas.berrios@sansano.usm.cl,
alejandro.vilches@sansano.usm.cl

Fecha de Entrega: 18 de octubre, 2019
Plazo máximo de entrega: 3 días.

Reglas del Juego

La presente tarea debe hacerse en grupos de 3 personas. Toda excepción a esta regla debe ser conversada con los ayudantes **antes** de comenzar la tarea. No se permiten de ninguna manera grupos de más de 3 personas. Las tareas deben compilarse en los computadores que se encuentran en el laboratorio B-032 (LDS). Deben usarse los lenguajes de programación C o C++. Se recomienda compilar en el terminal usando `gcc archivo.c -o output -Wall` (en el caso de lenguaje C) o `g++ archivo.cpp -o output -Wall` (en el caso de C++). Si usa C++, está permitido emplear estructuras de datos de la `stl`, en caso de ser necesarias.

Convex Hull

El *convex hull* de un conjunto S de n puntos en el plano es el polígono convexo de área mínima que contiene a todo el conjunto de puntos (ver el ejemplo de la Figura 1). Éste es un problema relevante para muchas aplicaciones.

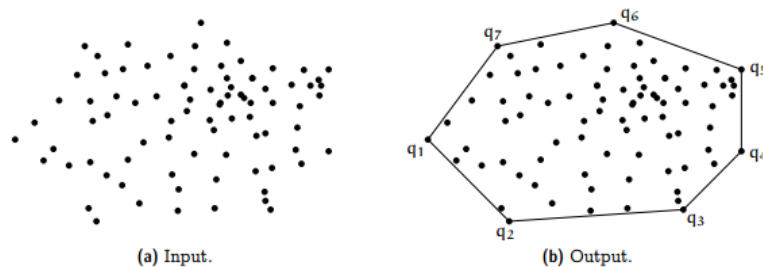


Figura 1: Convex Hull dado un Input (a) de una serie de puntos

Este problema puede ser abordado de diferentes formas, por lo que existen diversos algoritmos que lo resuelven de forma eficiente. Sin embargo, antes de estudiar dichas soluciones, estudiaremos cotas inferiores asociadas a este problema.

Para encontrar dicha cota inferior, en esta tarea usaremos el enfoque de *reducción de un problema a otro*. Un problema P se reduce a otro problema Q si se demuestra que cualquier algoritmo que resuelve a Q puede ser usado para resolver P . La demostración consiste, generalmente, en mostrar cómo la entrada al problema

P debe ser transformada a una entrada para el problema Q , y luego mostrando cómo la respuesta producida para el problema Q puede usarse para responder al problema P .

Después de demostrar que un problema P se reduce a un problema Q , cualquier cota inferior demostrada para P también aplica al problema Q . De otra manera, si la cota inferior de Q fuera menor a la de P , la reducción permitiría resolver a P en menor tiempo que lo indicado por su cota inferior. Esto sería, sin dudas, una contradicción.

Objetivo de la Tarea

Demuestre que el problema convex hull tiene cota inferior $\Omega(n \log n)$ en el modelo de comparaciones. Para lograrlo, reduzca el problema de ordenamiento al problema de convex hull. Escriba un informe usando L^AT_EX. Este informe debe ser de a lo más una plana. Procure ser conciso y claro con sus argumentos y mantener la generalidad del problema.

Hint: para la demostración, considere que debe mostrar cómo transformar el conjunto de valores a ordenar (puede asumir que son números reales) en un conjunto de puntos, de tal manera que calcular el convex hull de dicho conjunto de puntos sea equivalente a ordenar el conjunto original.

Misión secreta en espacio no explorado

La organización secreta MIB llevará a cabo un lanzamiento masivo de naves espaciales con fin de explorar nuevas zonas de la galaxia. Sin embargo, la NASA no está contenta con esta operación, por lo que decidió instalar una red de satélites con el fin de tener un rastreo total del espacio aéreo y poner freno a las ambiciones de MIB. Considere que la red de satélites es modelada como un grafo, donde cada uno de los nodos corresponde a un satélite y los arcos conectan dos satélites que tienen comunicación directa en la red. Usted ha sido contratado por MIB como agente encubierto encargado de atacar esta red de satélites, generando cortes en las comunicaciones. El objetivo es encontrar las conexiones entre satélites de la red tal que, al ser eliminadas, generan dos o más subredes desconexas.

Formato de Entrada

La entrada se hará a través de la entrada estándar (`stdin`). Los datos de entrada consisten en una cantidad indefinida de casos de prueba finalizados con EOF. Cada caso de prueba corresponde a una posible red de satélites que ha creado la NASA, comenzando con una línea n que indica la cantidad de satélites que posee la red, seguido de n líneas donde el primer valor x_i corresponde al número del nodo (satélite), seguido de k valores ($0 \leq k \leq n - 1$), separados por espacio, que representan los nodos x_j (con $j \neq i$) con los que existe una conexión directa en la red.

Un ejemplo de entrada es el siguiente:

```
6
1 2 5 6
2 1 4
3 6 4
4 3 2
5 1
6 1 3
4
5 3 2
2 5 3
3 5 2
1
```

Hint: Para probar su programa de una mejor manera, ingrese los datos de entrada con el formato indicado en un archivo de texto (por ejemplo, el archivo `input.txt`). Luego, ejecute su programa desde la terminal, redirigiendo la entrada estándar como a continuación:

```
./tarea1 < input.txt
```

de esta manera evita tener que entrar los datos manualmente cada vez que prueba su programa, y evita errores.

Formato de Salida

La salida se hará a través de la salida estándar (`stdout`). Para cada caso de prueba se debe imprimir la o las conexiones que se deben atacar para poder generar una o mas subredes, en caso de no existir un corte posible se debe imprimir **No Existe Corte**. Cada caso de prueba debe estar separado por un salto de línea adicional. La salida correspondiente al ejemplo del enunciado es:

```
1 5
```

```
No Existe Corte.
```

Bonus por Eficiencia

| Lugar | Bonus |
|-------|-------|
| 1 | 30 |
| 2 | 20 |
| 3 | 10 |

Tabla 1: Bonus para las 3 tareas más rápidas.

Se darán bonus de a las 3 tareas que ejecuten más rápido y resuelvan correctamente el problemas. Dicho bonus se podrá usar en cualquiera de las tareas del curso, y será aplicado de manera automática procurando maximizar el promedio. Los bonus corresponden a los indicados en la Tabla 1.

Entrega de la Tarea

La entrega de la tarea debe realizarse enviando un archivo comprimido llamado

`tarea1-apellido1-apellido2-apellido3.tar.gz`

(reemplazando sus apellidos según corresponda) en el sitio Moodle (<https://moodle.inf.utfsm.cl>) del curso, a más tardar el día 18 de octubre, 2019, a las 23:55:00 hrs (Chile Continental), el cual contenga:

- Un archivo pdf con el informe de la resolución del problema 1.
- Los archivos con el código fuente necesarios para el funcionamiento de la tarea.
- `NOMBRES.txt`, Nombre y ROL de cada integrante del grupo.
- `README.txt`, Instrucciones de compilación en caso de ser necesarias.
- `Makefile`, Instrucciones para compilación automática.

Rúbrica de Evaluación

Para la corrección de la tarea se evaluarán los aspectos y descuentos estipulados en la Tabla 2 y 3. Cada ítem tendrá su puntaje del 1 al 100 y la nota final será el promedio de las ponderaciones de estos, que serán evaluados de acuerdo a los porcentajes indicados en las tablas correspondientes. Los descuentos son aplicados sobre la nota final.

| | |
|--|-------------|
| RESOLUCIÓN DEL PROBLEMA | 40 % |
| Argumentación válida | 50 % |
| Resolución final correcta | 50 % |
| FORMATO DE ENTREGA | 20 % |
| Respetar el formato de entrega (documento realizado en \LaTeX) | 50 % |
| Ortografía y redacción coherente | 50 % |
| TÉCNICAS DE RESOLUCIÓN | 40 % |
| Uso de estrategias de demostración correspondiente al curso y sugeridas | 100 % |

Tabla 2: Rúbrica de Evaluación con respecto al primer problema.

| | |
|---|-----------------------|
| MANEJO DE MEMORIA | 20 % |
| Uso correcto de memoria (Ej. No acceder a regiones no inicializadas) | 30 % |
| Uso eficiente de memoria | 30 % |
| Administración correcta de memoria (<code>malloc</code> , <code>realloc</code> , <code>free</code>) | 40 % |
| FORMATO I/O | 10 % |
| Uso correcto de la entrada y salida estándar | 50 % |
| Lectura e impresión en el formato solicitado | 50 % |
| TÉCNICA DE PROGRAMACIÓN | 70 % |
| Uso de la estrategia de programación solicitada en el enunciado | 25 % |
| Eficiencia del algoritmo implementado | 50 % |
| Uso de las estructuras de datos adecuadas para resolver el problema | 25 % |
| ÍTEM | DESCUENTO |
| No entregar en <code>.tar.gz</code> | -15 |
| Código fuente está desordenado | -15 |
| No incluir los archivos de documentación solicitados | -15 |
| Copiar (desde algún sitio web o compañero) ¹ | -100 |
| Tarea no compila | -100 |
| Tarea produce <i>segmentation fault</i> para cualquier caso | -100 |
| Atraso | -20 por día |
| <i>Warnings</i> de compilación | -5 por <i>warning</i> |

Tabla 3: Rúbrica de Evaluación con respecto al segundo problema.

¹El incidente será reportado con el jefe de carrera.