



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2133 — Estructuras de Datos y Algoritmos  
Segundo semestre 2017

## Tarea 4

**Fecha de entrega:** Jueves 30 de Diciembre a las 23:59

### Objetivos

- Construir un grafo para representar un problema de manera eficiente.
- Precomputar múltiples instancias de búsqueda sobre un grafo.

### Introducción

Los investigadores del centro de estudios biológicos *DCCUmbrella* están desarrollando un nuevo “medicamento” inteligente, el cual promete poder dirigirse de manera directa a determinadas células que serán su objetivo. Para esto es que te han contactado; dado el mapa de las células y un par de células, el virus debe poder determinar la trayectoria más eficiente para llegar de una célula a otra. En particular, interesa probar muchos pares de rutas para un mismo mapa de células, por lo que lo ideal sería precomputar esa información para poder responder las consultas en el menor tiempo posible.

### Mapa de Células

La primera parte del problema es construir un grafo a partir del mapa de células. El mapa está estructurado de la siguiente manera dentro de una imagen, como se muestra en la Figura 1:

- Cada célula es un grupo de píxeles contiguos de un mismo color.
- Dentro de cada grupo existe un píxel negro, que representa el núcleo de esa célula.

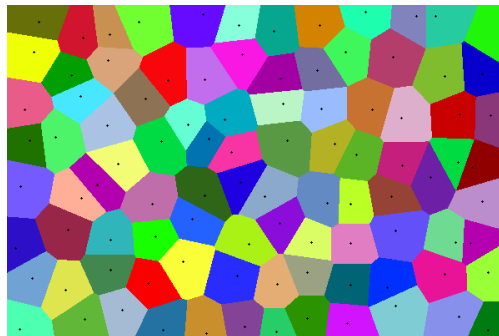


Figura 1: Un mapa de células. Ojo: Los núcleos están más grandes para ser más visibles

El grafo construido debe reflejar las siguientes restricciones:

- El virus sólo puede moverse a células vecinas a la célula en la que está, por lo que esas deben ser sus únicos vecinos en el grafo.
- El virus se mueve entre los núcleos de las células, por lo que la distancia entre núcleos es el costo de moverse de una célula a otra.

Se recomienda usar **DFS** y **Conjuntos Disjuntos** para identificar las distintas células y poder conectarlas en el grafo.

## Trazado de Rutas

La idea es poner a prueba al virus en una multitud de escenarios, por lo que una vez construido el grafo deberás entregar la ruta más corta entre cada par de células que se te consulte.

La idea es que precomputes las rutas más cortas entre todos los pares de nodos usando el algoritmo de **Floyd-Warshall**.

## Input

Tu programa debe recibir los siguientes parámetros:

1. La ruta al archivo de imagen con el mapa de células.
2. La ruta al archivo de consultas.
3. La ruta del archivo donde guardarás el grafo del mapa.
4. La ruta del archivo donde responderás las consultas.

de la siguiente manera:

```
./virustest <map> <queries> <graph> <routes>
```

## Mapa de Células

El mapa de células es simplemente una imagen PNG. Se espera que la abras usando las funciones de la librería provista. Asegurate de revisarla en `imgelib/imgelib.h`.

Esta expresará el mapa simplemente como una matriz de números, donde 0 corresponde al color negro, y los demás numeros entre 1 y 255 corresponden a los distintos colores de la imagen.

Como cada célula tiene un color único, ese número se usará como identificador de la célula.

## Archivo de Consultas

El archivo de consultas está estructurado de la siguiente manera:

- La primera línea contiene la cantidad  $k$  de consultas.
- Las siguientes  $k$  líneas contienen consultas, en el formato de dos números,  $i$   $j$ , donde  $i$  es la célula inicial y  $j$  es la célula final de la ruta.

## Output

El formato de los archivos de output es como sigue:

### Archivo de Grafo

Debe tener una línea por célula, en el formato

`C V E1 ... EV`

Donde **C** es el número correspondiente a la célula, **V** es la cantidad de vecinos, y los siguientes **V** elementos “E” son aristas, en el formato **B D**, donde **B** es el número correspondiente al vecino, y **D** es la distancia euclidiana del núcleo de la célula actual y el núcleo de ese vecino.

Por ejemplo, si la célula 4 tiene como vecinos a las células 2, 5 y 8, su línea se vería como sigue:

`4 3 2 50.800000 5 120.953002 8 70.333333`

### Archivo de Rutas

Debe tener una línea correspondiente a cada consulta, en el formato

`N i C1 ... CN j`

Donde **N** es la cantidad de células de la ruta, sin contar la célula inicial **i** y la célula final **j**. Las **N** células entre la inicial y la final corresponden a la ruta más corta para ir de **i** a **j**.

Por ejemplo, si lo más rápido para ir de la célula 5 a la 17 es pasar por la 8 y la 12, la consulta `5 17` tendría como respuesta la línea correspondiente

`2 5 8 12 17`

## Evaluación

Tu código será evaluado con diferentes tests generados aleatoriamente.

En esta tarea no habrá informe, por lo tanto la nota de tu tarea va a corresponder:

- 50% a que tu código genere el grafo correctamente.
- 50% a que las rutas calculadas sean correctas.

## Entrega

- **Lugar:** GIT - Repositorio asignado (asegúrate de seguir la estructura inicial de éste).
  - En la carpeta *Programa* debe encontrarse el código.
  - Se recogerá el estado en la rama *master*.
- **Hora:** 1 minuto pasadas las 23:59 del día de la entrega.
- Se espera que el código compile con `make` dentro de la carpeta *Programa* y genere un ejecutable de nombre `virustest` en esa misma carpeta.
- No se permitirán entregas atrasadas.