



CI-1221 Estructuras de Datos y Análisis de Algoritmos

II ciclo de 2017

III Tarea Programada

Fechas límite de entrega:

Parte I: lunes 13 de noviembre

Parte II: jueves 30 de noviembre

Descripción del problema

La Municipalidad de San José ha contratado a los excelentes estudiantes del grupo 2 del curso CI-1221 para que hagan una aplicación que mejore la seguridad de los conductores cuando los semáforos se encuentran desactivados (usualmente de 10 p.m. a 5 a.m). Para esto la aplicación debe diseñar una ruta que minimice la cantidad de veces que el conductor se detiene, ya sea por altos o por reductores de velocidad (también conocidos como *muertos*). Se escogió como plan piloto el distrito Catedral, cuyo mapa se muestra en la figura 1. Su tarea es escribir algoritmos que usen las técnicas de búsqueda exhaustiva, programación dinámica y, si es posible, algoritmos ávidos, que, dada una intersección de salida y una de llegada, encuentren la ruta óptima. Para reducir el tiempo de ejecución del algoritmo, la región de búsqueda se circunscribe al *rectángulo recubridor mínimo* formado por las dos intersecciones, es decir, las calles o transversales que se visiten deben estar comprendidas entre las calles o transversales de estas dos intersecciones (incluyendo estas) y las avenidas o diagonales que se visiten deben estar comprendidas entre las avenidas o diagonales de estas dos intersecciones (incluyendo estas), esto en cuanto a su numeración.¹ Para esto se debe considerar que en Costa Rica las calles en el sector este de las ciudades tienen una numeración impar y en el sector oeste par, y que las avenidas en el sector norte tienen una numeración impar y en el sur par.

¹Las *transversales* se originan a partir de un desvío hacia la izquierda entre 30° y 60° de una calle, recorrida en el sentido norte-sur (aunque los vehículos tampoco tengan derecho de vía en esa dirección), y heredan el número de la calle. Las *diagonales* se originan también a partir de un desvío a la izquierda de magnitud 30° a 60° pero a partir de una avenida recorrida en el sentido este-oeste (aunque los vehículos tampoco tengan derecho de vía) y también heredan el número de la avenida. Las transversales y diagonales en la vecindad de estas se definen continuando con el esquema de numeración de las calles y avenidas, respectivamente.

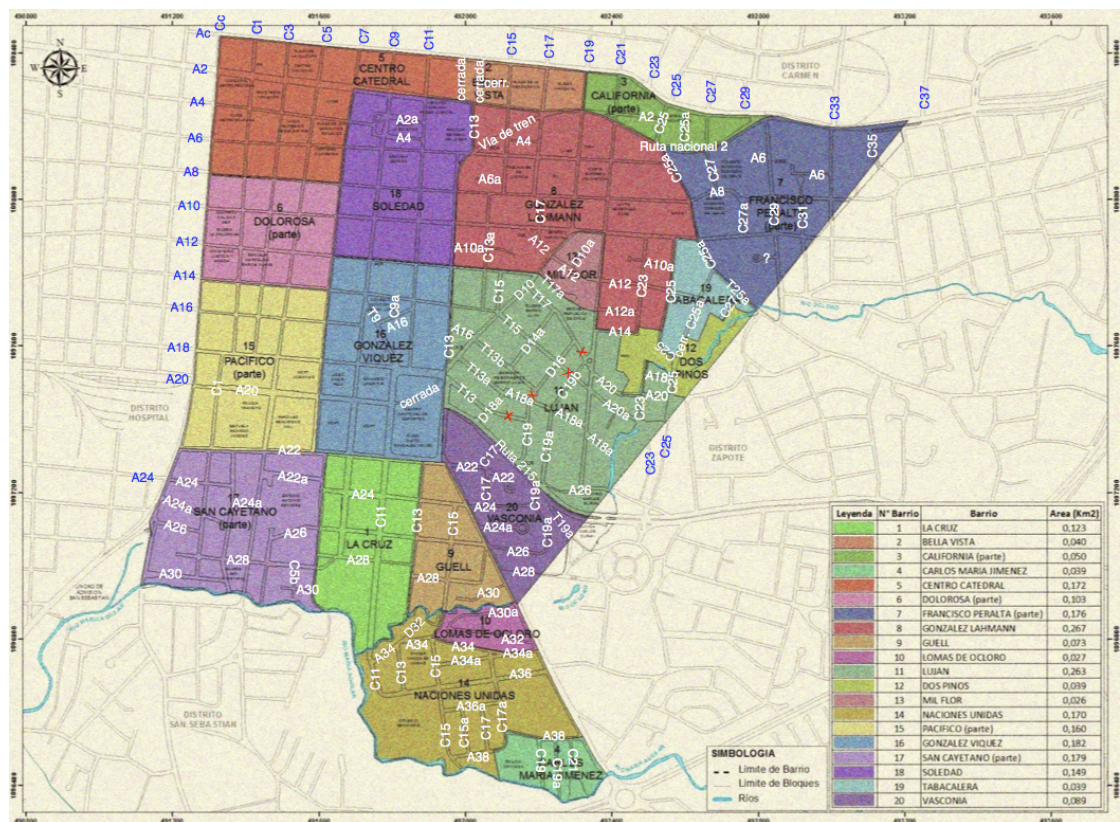


Figura 1: Mapa del distrito Catedral.

Los algoritmos deben basarse en la información recopilada por los y las estudiantes en la tarea 2, como preludeo a la tarea 3, y que está disponible en la carpeta <https://drive.google.com/drive/folders/OB6goUb7wEt0La21DRzFGNXM3U1k>. La salida del algoritmo debe ser una serie de instrucciones (en pantalla) que el *humano promedio* pueda entender para llegar de la salida al destino. La información sobre cada intersección visitada debe ser acompañada de la cantidad de altos y reductores de velocidad enfrentados (de forma combinada) para llegar a esa intersección (usualmente uno o ningún altos y varios o ningún reductor de velocidad). Debe suministrarse además la cantidad combinada de altos y reductores enfrentados a lo largo de todo el camino. Para esto se debe ignorar cualquier alto que haya en la intersección de salida (asuma que se empieza del centro de la intersección y, por tanto, cualquier alto que existiera para llegar ahí ya fue enfrentado hace unos cinco metros).

Conformación de grupos

Esta tarea se puede realizar en grupos de dos personas o de forma individual.

Entregables de la tarea

La tarea tiene dos entregas. La primera es un reporte preliminar que incluye la formulación y resultados obtenidos con el método de búsqueda exhaustiva (parte I). La segunda es un reporte final que incluye tanto este método como los de programación dinámica y algoritmos ávidos (parte II). Los reportes deben tener las secciones de un artículo científico estándar: título, resumen, introducción, metodología, resultados, conclusiones y bibliografía, aunque los nombres pueden variar a conveniencia. El reporte debe seguir el formato de la revista *Transacciones* del IEEE. Las especificaciones del formato y las plantillas para L^AT_EX y Word están disponibles en la página https://www.ieee.org/publications_standards/publications/authors/author_templates.html. LyX provee también una plantilla con este formato. (Vaya al menú Archivo ▸ Nuevo desde plantilla y seleccione IEEEtran-Journal.lyx).

Para obtener puntos extra, se puede adjuntar a la segunda entrega el formulario «Lista de chequeo de estilo - Tarea 3», en el que se reporta la adherencia a importantes reglas comunes de estilo. Eso sí, para poder afirmar que cumplió una regla, debe haberla cumplido en al menos un 90 % de los casos. *Si afirma haber cumplido una regla y no la cumplió en el 90 % de los casos, no solo no se le darán los puntos sino que además se le castigará con la misma cantidad de puntos.*

Forma de entrega

La tarea debe entregarse antes de la fecha y hora especificadas, por medio de la plataforma MEDIACIÓN VIRTUAL. La tarea se puede dejar en estado «borrador» (no es necesario «enviarla a revisión»). Basta con enviar una versión por grupo (no es necesario

que cada miembro del grupo la envíe; con solo uno que lo haga es suficiente). Es responsabilidad del estudiante verificar que la plataforma haya recibido la tarea y que esté intacta (bajando la tarea y verificando su integridad, especialmente si está comprimida, p. ej., en formato ZIP). En caso de que haya múltiples entregas, se considerará solamente la *última*, y si esta se entregó después de la fecha y hora límites, se aplicará la penalización acordada en la CARTA AL ESTUDIANTE: *la calificación recibida por tareas tardías no será más alta que la calificación recibida por cualquiera de los estudiantes que haya entregado la tarea a tiempo*. Si tiene problemas para subir la tarea y el plazo está cerca de cumplirse, envíela a los asistentes, con copia al profesor. Favor poner en el «asunto» lo siguiente: *Tarea de Estructuras de Datos*. En caso de que se reciban múltiples versiones por correo electrónico, se revisará solamente la *primera* enviada. La política de penalización por entrega tardía mencionada en el párrafo anterior también aplica a entregas hechas por correo electrónico.

Parte I.

1. Búsqueda exhaustiva (50 pts.)

1. Encuentre una representación vectorial de la solución al problema, indicando claramente el significado de cada una de sus entradas. [2 pts.]
2. Determine el espacio E en el que habita la solución, definiendo claramente el (los) conjunto(s) a el (los) que pertenece(n) cada una de las entradas. [2 pts.]
3. Determine la cardinalidad del espacio. [2 pts.]
4. Si es posible, acote el espacio mediante una restricción del tipo $E' = \{s \in E : \text{restricción en } s\}$. [5 pts.]
5. Escriba un algoritmo en C++ que explore todas las soluciones candidatas en E o E' y devuelva una solución óptima [30 pts.]. El puntaje obtenido en esta sección es

$$\frac{30}{k} \sum_{i=1}^k \frac{O_i}{O'_i}, \quad (1)$$

donde k es la cantidad de casos probados en la revisión, O'_i es la cantidad de obstáculos (altos y reductores) que la ruta hallada por el algoritmo enfrentó y O_i es la cantidad mínima de obstáculos que se pueden enfrentar. (Esto último lo conocen los asistentes y el profesor, pero los guardan en extremo secreto).

6. Determine una cota asintótica para el tiempo de ejecución del algoritmo. [2 pts.]

7. Ejecute el algoritmo con cada uno de los casos de prueba dados. Si el tiempo de ejecución excede 10 minutos para una cierta ruta, puede abortarla. Reporte en un cuadro el tiempo de ejecución de cada una de las ejecuciones y grafique los tiempos como función de la longitud esperada del semiperímetro del rectángulo recubridor mínimo.² [5 pts.]
8. Indique si el crecimiento de los tiempos fue el esperado y explique por qué. [2 pts.]

Parte II.

2. Programación dinámica (50 pts.)

1. Determine si el problema se puede resolver a partir de soluciones a subproblemas, y si es así, describa cómo construir tal solución. [2 pts.]
2. Determine un oráculo (describiendo claramente su significado y el de sus argumentos) [3 pts.], el valor objetivo [2 pts.], y los pasos base [2 pts.] y recursivo [8 pts.] que permiten construirlo.
3. Basado en su respuesta al punto anterior, escriba un algoritmo que resuelva el problema [20 pts.]. El puntaje obtenido en esta sección será

$$\frac{20}{k} \sum_{i=1}^k \frac{O_i}{O'_i}, \quad (2)$$

donde k es la cantidad de casos probados en la revisión, O'_i es la cantidad de obstáculos (altos y reductores) que la ruta hallada por el algoritmo enfrentó y O_i es la cantidad mínima de obstáculos que se pueden enfrentar.

4. Determine una cota asintótica para el tiempo de ejecución del algoritmo. [3 pts.]
5. Ejecute el algoritmo usando los casos de prueba suministrados. Indique si las soluciones obtenidas producen el mismo número de obstáculos enfrentados que las soluciones obtenidas mediante búsqueda exhaustiva [3 pts.]. Grafique los tiempos de ejecución vs. la longitud esperada del semiperímetro del rectángulo recubridor mínimo, representando los tiempos en escala logarítmica, y agregue al gráfico la curva correspondiente a los tiempos producidos por el algoritmo de búsqueda exhaustiva [4 pts.]. Compare las curvas y diga si la relación entre los tiempos fue la esperada [3 pts.]

²El *semiperímetro* de un polígono es la mitad de su perímetro. En Costa Rica, los lados de las cuerdas tienen en promedio una longitud de 100 metros.

3. Algoritmos ávidos (5–35 pts.)

1. Determine si es posible resolver el problema mediante un algoritmo ávido cuyo tiempo de ejecución sea asintóticamente mejor que el de programación dinámica. Explique su respuesta. [5 pts.]
2. Si su respuesta al punto anterior fue afirmativa:
 - a) Escriba un algoritmo ávido que resuelva el problema [10 pts.]. El puntaje obtenido en esta sección será

$$\frac{20}{k} \sum_{i=1}^k \frac{O_i}{O'_i}, \quad (3)$$

donde k es la cantidad de casos probados en la revisión, O'_i es la cantidad de obstáculos (altos y reductores) que la ruta hallada por el algoritmo enfrentó y O_i es la cantidad mínima de obstáculos que se pueden enfrentar.

- b) Use el «método de la transformación» para demostrar que la estrategia es correcta. [7 pts.]
 - c) Determine una cota asintótica para el tiempo de ejecución del algoritmo. [3 pts.]
 - d) Ejecute el algoritmo usando los datos de prueba suministrados. Indique si las soluciones obtenidas producen el mismo número de obstáculos enfrentados que los algoritmos anteriores [3 pts.]. Grafique los tiempos de ejecución vs. la longitud esperada del semiperímetro del rectángulo recubridor mínimo, representando los tiempos en escala logarítmica, y agregue al gráfico la curva correspondiente a los tiempos producidos por el algoritmo de búsqueda exhaustiva y programación dinámica [4 pts.]. Compare las curvas y diga si la relación entre los tiempos fue la esperada [3 pts.]