



**Departamento de  
Informática e Ingeniería  
de Sistemas**  
**Universidad Zaragoza**

**Prácticas de Algoritmia para problemas difíciles  
Especialidad en Computación, grado en Ingeniería Informática**

**Curso 2018-2019**

Universidad de Zaragoza  
Escuela de Ingeniería y Arquitectura  
Departamento de Informática e Ingeniería de Sistemas  
Area de Lenguajes y Sistemas Informáticos

13 de noviembre de 2018

# Organización general de las prácticas.

Se formarán equipos de una o dos personas. Si una de las dos abandona la asignatura, la otra deberá terminar en solitario. Se debe enviar un email con el nombre de los miembros del equipo antes del 15 de noviembre de 2018.

Las prácticas se realizarán en el computador **hendrix**. Excepcionalmente pueden aceptarse otras alternativas siempre que puedan ejecutarse directamente en Cygwin o en Windows 10.

El lenguaje para la implementación puede ser Java o cualquier otro elegido por el alumno (incluyendo en la documentación todos los detalles necesarios sobre el mismo: versión, compilador, etc).

La entrega de la práctica X ( $X = 1, 2, \dots$ ) se realizará en **hendrix** mediante la ejecución de `someter apd_18 practicaX.tar`, siendo la fecha límite la indicada al final de cada enunciado. Además, habrá que concertar una cita con la profesora para explicar las prácticas: descripción general del programa, elaboración, funcionamiento, demostración, etc.

**Nota:** En caso de que el fichero resultante tenga un tamaño mayor de 1024 KB deberás dividir el fichero `practicaX.tar` en varios ficheros de tamaño 1024KB con la orden de linux

```
split -b 1m practicaX.tar practicaX.tar.
```

Se crearán los ficheros `practicaX.tar.aa`, `practicaX.tar.ab`, ..., cada uno de tamaño máximo 1024KB.

El fichero `practicaX.tar` contendrá **un directorio denominado `practicaX`** con los ficheros necesarios incluyendo:

- Una memoria en formato PDF en la que se incluya un informe completo con al menos:
  - Una descripción de qué se ha implementado y usando qué fuentes (web, libros, artículos, código propio, etc). Cualquier decisión tomada sobre implementación y sus razones. Si hay algo que decir sobre el lenguaje de programación utilizado.
  - Cómo se ha repartido el trabajo entre los dos integrantes del equipo.
  - Qué pruebas se han hecho, de dónde se han sacado los datos, resultados obtenidos (incluyendo eficiencia).

- Descripción general del programa en fichero de texto: cómo está organizado, qué se puede y qué no se puede hacer (tiene que llamarse **LEEME**). Contendrá en sus primeras líneas la lista de integrantes del grupo, con el siguiente formato:

```
Apellido1 Apellido2, Nombre [tab] correo@electronico [tab] login en  
hendrix
```

```
Apellido1 Apellido2, Nombre [tab] correo@electronico [tab] login en  
hendrix
```

Donde [tab] representa el carácter tabulador.

- Listados del código debidamente comentados y dispuestos para ser compilados y utilizados.
- Un programa para el shell `ejecutarX.sh` que automatice la compilación y ejecución de algunos casos de prueba para los programas entregados. Deberá funcionar en hendrix (excepcionalmente en Cygwin/Windows 10) .
- Los ficheros auxiliares de entrada necesarios para ejecutar las pruebas del punto anterior.
- Los ficheros de otras pruebas realizadas.

En la calificación se tendrán en cuenta los siguientes aspectos: documentación, funcionamiento e implementación.

El diseño ha de ser modular, basado en el uso de tipos abstractos de datos, con todas las funciones correctamente especificadas.

Las reglas generales de tratamiento de casos de plagio de la asignatura se aplicarán, en particular, a todas las prácticas.

# Práctica 1

Te está costando elegir entre los dos principales servicios de mensajería que hay en tu ciudad.

Tienen el mismo precio, el servicio al cliente tiene coste muy bajo y usan la misma estrategia de reparto, excepto que empiezan en distinto almacén.

Su estrategia es la siguiente, un conductor sale de un almacén que está directamente conectado a una intersección. Cuando un conductor llega a una intersección  $I$  se elige aleatoriamente una carretera, dada una distribución de probabilidad conocida de las carreteras que salen de  $I$ . Cada intersección está conectada con carreteras que van al menos a una intersección distinta. Lleva un tiempo fijo recorrer una carretera. Si un conductor llega a la intersección donde vives te entrega el paquete (nunca te vas de casa).

Te gustaría elegir el servicio de mensajería que tiene el menor tiempo estimado de reparto.

## 1.1. Entrada

La primera línea contiene 5 enteros,  $N M C A B$ , donde  $3 \leq N \leq 300$  es el número de intersecciones,  $\frac{N}{2} \leq M \leq \frac{N(N-1)}{2}$  es el número de carreteras,  $0 \leq C < N$  es la intersección en la que vives, y  $0 \leq A, B < N$  son los almacenes de las dos mensajerías.

A continuación  $M$  líneas describen cada carretera.

Una carretera se describe con 5 números,  $u, v, t_{uv}, p_{uv}, p_{vu}$  donde  $u$  y  $v$  son las intersecciones que conecta la carretera,  $t_{uv}$  es un entero describiendo el tiempo en minutos que cuesta recorrer la carretera,  $p_{uv}$  es la probabilidad de elegir viajar de  $u$  a  $v$  si estás en la intersección  $u$ , mientras que  $p_{vu}$  es la probabilidad de elegir viajar de  $v$  a  $u$  si estás en la intersección  $v$ .

Todas las probabilidades  $p_{uv}$  con  $u \neq C$  cumplen  $0 < p_{uv} \leq 1$ .

Para cada intersección  $u$  la suma de probabilidades  $p_{uv}$  es 1, excepto para  $C$ , donde el conductor entrega el paquete, aquí la probabilidad de irse es 0.

## Resultado

El resultado es el tiempo de entrega estimado para cada mensajería. Si tu casa no es alcanzable para una mensajería el resultado será un mensaje “Intentamos entregar tu paquete, pero no estabas”.

### 1.2. ¿Qué hay que hacer?

1. Hasta el **20 de diciembre** (inclusive) cada equipo puede entregar cuantos conjuntos de datos de prueba considere, con el objetivo de cubrir el mayor número de casos posible y los tamaños que se pretendan resolver. Estas entregas puntuarán hasta el 25 % de la nota de la práctica y deberán someterse en `hendrix` mediante la ejecución de `someter apd_18 datosPractica1X.tar`, con  $X = 1, 2, \dots$ . Además de los datos se debe incluir un breve fichero `.txt` que explique cómo se han generado. Posteriormente estos datos estarán disponibles para todos los equipos en `/misc/practicas/apd`
2. Diseñar e implementar un algoritmo probabilista que simule atravesar el mapa desde cada almacén hasta tu casa calculando el tiempo. Se estimará el tiempo de entrega como media de las simulaciones.
3. Probar el algoritmo sobre cada conjunto de prueba.

## Créditos

Esta práctica está basada en el artículo “Canadians should travel randomly” de Demaine et al. 2014 <https://dspace.mit.edu/openaccess-disseminate/1721.1/99995> y en material del curso de Algoritmos Avanzados de la Universidad de Lund (Suecia), coordinado por el profesor Thore Husfeldt <http://cs.lth.se/edan55/> (Nota: Este curso es bastante más avanzado que el nuestro.)

### 1.3. Entrega

Deberá entregarse **hasta el 7 de enero de 2019**.