



2018/1

2017/1

2016/2

Trabajo Práctico n.º 3

Teoría de Algoritmos 1 - 1c 2018 Trabajo Práctico 3

Lineamientos básicos

- El trabajo se realizará en grupos de cuatro personas (excepcionalmente de tres).
- La fecha de entrega es el lunes 18 de Junio de 2018. Se debe entregar en el horario de clase en papel (informe + código en monoespacio), más una entrega en digital de código (.zip) e informe (.pdf) al correo de entregas del curso: tps.7529rw@gmail.com.
- El lenguaje de implementación es libre, pero se debe comunicar por correo en caso de no ser uno de: C, C++, Python, Java, JavaScript, Ruby, Go, Rust, Swift, Scala, Haskell, OCaml, Clojure, D, Lua, Elixir.
- Incluir en el informe los requisitos y procedimientos para su compilación y ejecución.

Parte 1: Un juego de batalla naval





En un océano remoto dos bandos pelean entre ellos. El bando "A" tiene un conjunto de barcos e intenta avanzar para dominar más territorio. El bando "B" tiene un conjunto de lanzadoras de misiles con los que intenta neutralizar a los rivales.

Podemos modelizar el juego como:

- El juego se realiza por turnos.
- El bando "A" tiene X barcos. Cada uno de ellos tiene V_i puntos de vida.
- El bando "B" tiene Y lanzaderas de misiles. Cada una tiene un alcance global (es decir que llega a cualquier lugar que desee).
- El tablero de juego tiene forma de grilla. Con igual cantidad de filas que barcos iniciales. Las columnas son ilimitadas (tantas como se necesiten en el juego). A efectos programáticos se puede considerar que al llegar al final de la grilla se "teletransporta" al inicio de la misma.
- Cada barco se ubica al inicio de una fila y se desplaza por turno en 1 posición (a la columna siguiente).
- Cada lanzadera de misiles puede disparar 1 misil por turno.
- El daño que puede realizar un misil está únicamente ligado a la grilla en la que se encuentra el barco en ese momento.
- Los valores de daño están predefinidos en la grilla al inicio del juego, no se modifican y son conocidos por "A".
- Por cada turno que pasa el equipo "B" recibe un punto por cada barco que siga en el juego.
- Los barcos permanecen en el juego y avanzan siempre que conserven puntos de vida.
- Los puntos de vida y la cantidad de daño son valores enteros positivos.
- El objetivo de "B" es minimizar la cantidad de puntos logrados por "A".

Tenemos 2 amigos que usarán el juego. "Greedy" y "Dinámico". Jugarán partidas entre ellos alternando entre el bando "A" y "B". Siempre el juego se realizará de a pares conservando la misma disposición de tablero y fichas (lanzaderas y barcos).

1. Diseñar para "Greedy" una estrategia greedy y para "Dinámico" una estrategia mediante programación dinámica. Presentar el pseudocódigo. Analizar sus complejidades.
2. Programar el juego permitiendo tomar diferentes grillas, lanzaderas y barcos. debe poder usarse cualquiera de las dos estrategias realizadas en el punto 1.

3. Determinar si la solución greedy es óptima o si no lo es bajo qué condiciones puede serlo.
4. (opcional) Realice una interfaz gráfica para visualizar la evolución del juego
5. Imagine que el jugador "A" tiene la posibilidad de seleccionar para cada barco la columna inicial donde se ubicará su barco. Diseñe un algoritmo que le permita aumentar la cantidad a lograr. Analice su complejidad

Grilla:

Se deberá contar con un archivo con la definición del tablero de juego. El mismo deberá tener el siguiente formato:

- Un renglón por fila de grilla
- El primero número de cada fila es la vida del barco que se movera en ella.
- Seguido de números enteros separados por espacio para cada daño de recibir un misil en la columna

Ejemplo:

```
800 20 40 10 120
300 80 30 90 0
```

El número de filas determinará la cantidad de barcos del jugador "A". Por parámetro del programa se le debe pasar la grilla a utilizar, la estrategia y la cantidad de lanzaderas disponibles.

La salida del programa debe mostrar por cada turno:

- Número de turno
- Cantidad de barcos disponibles
- Vida restante de cada barco y daño potencial según celda en la que se encuentra
- Objetivos seleccionados por las lanzaderas.
- Puntos acumulados

A la finalización debe indicar

- Turnos totales
- Puntos acumulados.

Parte 2: Sabotaje!



Son agentes secretos de la organización C.O.N.T.R.O.L y reciben una información anónima que los pone en alerta: La organización K.A.O.S planea sabotear la red secreta de transporte de información. Hace unos días se reportó el robo de una copia del mapa maestro de la red. Dan por supuesto que el robo fue de sus rivales y que ellos conocen las debilidades de su red. Siendo que únicamente tienen personal suficiente para vigilar 2 ejes de su red, que únicamente se pueden sabotear ejes (y no vertices) y que los saboteadores quieren hacer el máximo daño posible:

1. Diseñe un algoritmo genérico que funcione con cualquier tipo de red que determine qué ejes vigilar. Preséntelo con un pseudocódigo, explique su funcionamiento y si el mismo además es óptimo.
2. Analice la complejidad del mismo. De igual manera analice si utiliza alguna reducción.
3. Programe el algoritmo.
4. Si es necesario programe el algoritmo para determinar el flujo máximo. Utilícelo para calcular el de la red antes y después de los posibles sabotajes.
5. En una situación real existirán varias fuentes y varios sumideros. Explique y analice una forma de resolver el mismo problema en este caso.

Red Secreta:

El mapa de la red se debe obtener de un archivo llamado "redsecreta.map" con el siguiente formato:

- Cada línea del archivo corresponde a un eje de la red que une 2 vértices y su capacidad
- Los ejes están etiquetadas por números enteros

- La capacidad son números enteros.
- La fuente estará etiquetada como el número 0 (cero)
- El sumidero estará etiquetado como el número 1 (uno)

Aclaraciones generales:

En esta sección agregaremos aclaraciones y correcciones que surgen de consultas en la clase.

Una batalla naval:

- (CORRECCION) Se modifica en el enunciado
 - El objetivo de "A" es minimizar la cantidad de puntos logrados por "B"
- por:
- El objetivo de "B" es minimizar la cantidad de puntos logrados por "A"

Sabotaje:

- (ACLARACION) el sabotaje esperado es en una arista de la red
- (ACLARACION) Las capacidades de cada arista pueden o no ser diferentes.
- (CORRECCION) En la definicion del archivo "redsecreata.map deberia decir:

Los **vertices** están etiquetadas por números enteros

Ejemplo de linea del archivo:

0 4 80

Que significa "la fuente esta unida al vertice 4 y su capacidad es 80