

# Reconquistando la Tierra de Fantasía

En esta aventura, acompañarás a una raza de guerreros en su misión por reconquistar la tierra que le ha sido arrebatada en anteriores batallas. Estos guerreros han recobrado fuerzas y disponen de un ejército que consideran suficientemente poderoso para emprender esta epopeya.

De camino a la tierra deseada se encontrarán con diversos poblados. Algunos de los cuales son hostiles, a los que deberán vencer para poder continuar el camino. Otros poblados son aliados, y permitirán que las tropas descansen y la mitad de su población se sumarán a tu ejército.

Es importantísimo no desperdiciar recursos y, aún a riesgo de no encontrar poblados aliados, recorrer el camino más rápido entre su ubicación actual y la tierra destino: el tiempo apremia. Cada batalla o descanso, dura un día.

Se te buscó, hechicero del código, para que prepares unos conjuros algorítmicos que te permitan:

1. **predecir si esta misión es factible, y**
2. **en caso de serlo:**
  - **cuántos guerreros llegarían hasta el final del camino, y**
  - **en cuánto tiempo.**

## Datos para resolver el problema

### Razas

Un **Wrives** tiene una salud inicial de 108. Utiliza magia, y su rango de ataque es de 14 a 28 metros. Ocasiona un daño básico de 113 puntos. Cuando ataca, lo hace con 2 veces su daño, cada 2 ataques. Al recibir un ataque recibe 2 veces el daño, ya que no lleva armadura. Cuando descansa, medita, y como considera la violencia como algo malo, se rehúsa a atacar hasta que lo ataquen. Gracias a esto, aumenta su salud y su salud máxima en 50.

Una **Reralopes** tiene una salud inicial de 53. Utiliza una catapulta, y su rango de ataque es de 5 a 46 metros. Ocasiona un daño básico de 27 puntos. Cuando ataca, erra 2 de cada 4 ataques. Al recibir un ataque se desconcentra y sus ataques vuelven al valor normal inicial. Cuando descansa, se concentra y sus próximos 3 ataques (de esa unidad) dañan el doble del valor correspondiente.

Una **Radaiteran** tiene una salud inicial de 36. Utiliza shurikens, y su rango de ataque es de 17 a 41 metros. Ocasiona un daño básico de 56 puntos. Cuando ataca, lo hace cada vez con más

Trabajo Práctico - Algoritmos y Programación 2 - UNTreF  
Mayo - Junio de 2021

fuerza (3 de daño extra x la cantidad de ataques dados). Al recibir un ataque lo hace normalmente. Cuando descansa, no le sucede nada.

Un **Nortaichian** tiene una salud inicial de 66. Utiliza un arco, y su rango de ataque es de 16 a 22 metros. Ocasiona un daño básico de 18 puntos. Cuando ataca, se cura un 4 por ciento de su salud. Al recibir un ataque se enfurece y sus ataques multiplican por 2 su daño (dura 2 turnos propios). Cuando descansa, recupera toda su salud, pero se vuelve de piedra por 2 turnos (contiguos), lo que hace que no pueda atacar, pero reduce el daño entrante en 1/2.

## Batallas

Las batallas en la Tierra de Fantasía se realizan de una manera muy ordenada:

1. Se forman ambos ejércitos en línea. Nuestro ejército formará primero a las unidades aliadas, luego a las propias.
2. La unidad que haya quedado herida de la batalla anterior siempre será la última en recibir ataques.
3. Siempre comienza a atacar nuestro bando.
4. Se turnan ambos ejércitos para atacarse.
5. Al quedarse con la salud en cero, la unidad se desmaya y queda fuera de combate y no continúan la misión.
6. Termina el combate cuando un ejército se queda sin contendientes de pie.

## Poblados

Se suministrará un archivo con la información de los caminos que interconectan a los poblados, y los datos de dicho poblado. Por ejemplo:

```
4
1 100 Wrives propio
2 30 Reralopes aliado
3 40 Nortaichian enemigo
4 60 Nortaichian enemigo
1 -> 4
1 2 10
1 3 20
2 3 5
3 4 7
```

En ese archivo figura toda la información necesaria para la predicción:

- Una línea con la cantidad de pueblos (n, 4 en el ejemplo).

Trabajo Práctico - Algoritmos y Programación 2 - UNTreF  
Mayo - Junio de 2021

- n líneas autonumeradas, que representan cada pueblo, con el total de habitantes, la raza, y si es propio / aliado / enemigo. "Propio" será un único pueblo.
- Una línea que indica el pueblo inicial, y el pueblo final (1 -> 4).
- x líneas que indican el costo de trasladarse entre cada par de pueblos, siendo estos datos pueblo de origen, pueblo destino, costo del traslado en días.

## Condiciones

1. **Las razas deberán programarse utilizando la técnica de TDD.** Una opción es que a cada prueba escrita se le asigne un número consecutivo, que evidencie el desarrollo incremental. Por ejemplo, prueba001, prueba002, etc. A medida que avancen, es muy probable que terminen separando las pruebas en distintas clases, según qué raza se está probando. **Se verificará una cobertura mayor al 92%.**
2. **El camino más corto deberá calcularse utilizando un algoritmo de grafos apropiado.** El mismo deberá programarse en términos de un grafo y no de los terrenos, para poder ser reutilizado en futuros usos.
3. **El mapa es único para todo el problema,** y debe poder accederse a la misma instancia desde cualquier clase que lo requiera. Utilizar un patrón de diseño para este comportamiento.
4. **Tanto el ejército como la unidad individual deben poder tratarse de manera uniforme para el ataque, la recepción de golpes y el descanso.** Utilizar un patrón de diseño para este comportamiento.

## Puntos extra (opcional)

El grupo que, dada una configuración que resulte en derrota, pueda dar programáticamente una alternativa determinística que acabe con la victoria, teniendo el menor costo posible, con una complejidad computacional aceptable.

## Sobre las entregas

Cada equipo tiene asignado **un/a tutor/a**, el cual los acompañará durante el desarrollo del trabajo práctico, y será quien lo corrija. Es importante que aprovechen esta relación para evacuar dudas, discutir implementaciones, y coordinar las entregas. Todo esto debe ser coordinado con su tutor previamente, con suficiente antelación.

Habrà una **preentrega**, para confirmar que el rumbo que tomaron es el correcto, más allá de las consultas que puedan coordinarse. Esta preentrega deberá tener al menos el diagrama de clases del problema, para poder discutir la solución.

La **fecha de esta primera aproximación se coordinará con cada tutor**.

Para la entrega final, el 22 y 23 de Junio (dependiendo su día de cursada), se deberá disponer de los siguientes entregables:

- **Diagrama de clases** actualizado (.jpg)
- **Pruebas unitarias** del código. Dichas pruebas deberán validar *significativamente* los aspectos críticos del programa.
- El **código**:
  - debidamente comentado (lease: siempre y cuando se evidencie necesario y no sea redundante ni trivial),
  - debidamente formateado, y
  - sin errores de compilación.
- Un **main** que evidencie el funcionamiento del programa.
- Al menos **cuatro archivos de entrada** de prueba, *significativamente distintos*, para poder ejecutar distintas versiones del problema.
- Un **breve informe** (.pdf) que explique:
  - **decisiones de diseño** (cuáles aspectos del problema consideraron relevantes para el diseño y cuáles no, qué opciones de implementación evaluaron, cuál fue la alternativa elegida en cada caso, y por qué), **sin entrar en detalles de implementación**, y
  - **organización y distribución del trabajo**.

Opcionalmente, también puede agregarse cualquier otro aspecto que consideren relevante (Ej: explicar el funcionamiento general de la solución para un ejemplo concreto.).