Experimentación

Para el proceso de experimentación del problema se plantearon distintos escenarios de test para corroborar que el algoritmo propuesto funcionara correctamente y que la cota de complejidad encontrada y justificada en la sección anterior, en la práctica, se cumpliera.

Llamamos escenario de test a un conjunto de pruebas que si bien son distintas, comparten alguna similitud.

Por ejemplo, un escenario es aquel en el cual un participante puede dar saltos de una distancia k constante, y las distintas pruebas del escenario variarían en el tamaño del puente y en el estado de sus tablones.

Dado que el CPU de la computadora utilizada para tomar los tiempos no está atendiendo únicamente a nuestro proceso, realizar una sola vez cada prueba podría darnos valores que no son cercanos a los reales. Para minimizar este margen de error, a cada prueba de cada escenario se la hizo ejecutar un total de 10.000 veces y se tomó el mejor valor. Notar que tomar el mejor valor no es una mala decisión, ya que mientras más chico sea el valor, más cerca estamos del valor real de tiempo que toma el algoritmo para una instancia dada.

En cada prueba se tomaron métricas para la posterior evaluación del algoritmo en la práctica. Notar que la medición no contempla tiempos de entrada/salida de datos, sino que contempla solamente el núcleo del algoritmo.

Para cada escenario testeado, se hicieron gráficos 2D que permitan ver de una manera más clara los resultados obtenidos en las pruebas del mismo. Estos fueron realizados con el software QitPlot que la cátedra proveyó.

En cuanto a qué casos testear, decidimos testear los casos “border” y casos aleatorios, mezclando en todos los casos distintas variaciones de saltos posibles, tamaño de los puentes y estado de los tablones de los mismos.

Los casos "border", son aquellos que están en los extremos de las capacidades del algoritmo, es decir, el mejor caso que el algoritmo puede resolver, y el peor.

Para las instancias aleatorias, se diseñó un generador de estas, que dada una longitud, el estado de generaría los tablones se haría de forma aleatoria, es decir, dado un tablón t, en una prueba t puede ser válido y en otra no. Este generador es capaz de generar múltiples instancias aleatorias.

Para todos los casos, se eligió una precisión de hasta 0,0001 ms (milisegundos). De ser menor, la notamos como 0.

En todos los casos se pudo comprobar que la práctica refleja lo expuesto en incisos anteriores.

Escenario de mejor caso del algoritmo.

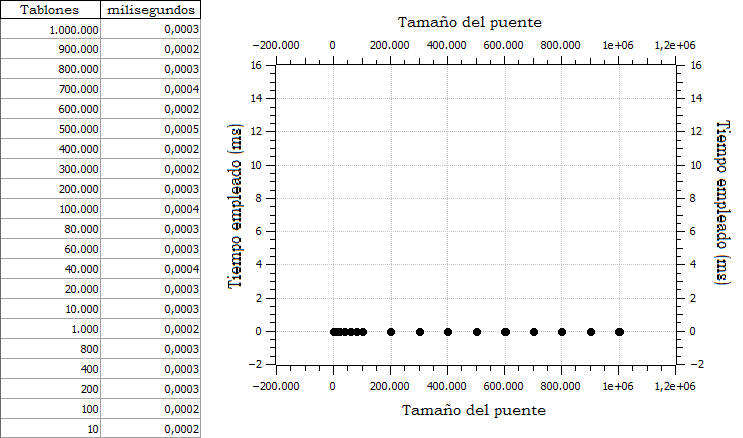
Para el algoritmo propuesto, el mejor caso de respuesta es el caso en el que la capacidad de salto del participante es superior a la longitud del puente, este caso tiene una complejidad de orden constante, es decir, el mejor caso es O(1).

Notar que para probar el mejor caso, la capacidad de salto no puede ser un parámetro variable, ya que es necesario que sea mayor que la cantidad de tablones.

Para evaluar el mejor caso, el generador de instancias, aplica como salto máximo, el tamaño del puente + 1. Por ende, para cada prueba realizada, si el puente contenía n tablones, el salto era de n+1.

Dicho esto, se generaron distintas instancias del algoritmo, con distintos tamaños, y con puentes de estado aleatorios, para verificar que el mejor caso no del estado de sus tablones, sino de que el salto sea mayor estricto que la cantidad de tablones.

Luego de realizar las pruebas pertinentes de este escenario, (y de realizar 10.000 veces cada una) como declaramos al inicio del insiso, mostramos los resultados obtenidos en el siguiente gráfico:



En la figura superior se puede apreciar que el tiempo de resolución es constante a pesar de variar el tamaño de entrada y el estado de los tablones (generados aleatoriamente).

Escenario de peor caso del algoritmo.

Para el algoritmo propuesto, el peor caso de respuesta, es decir, el caso que más tiempo demanda ejecutar, es aquel en el que el participante puede cruzar todo el puente (por lo que el algoritmo debe continuar hasta el final del mismo) y dando saltos mínimos de avance, es decir, saltos de longitud 1.

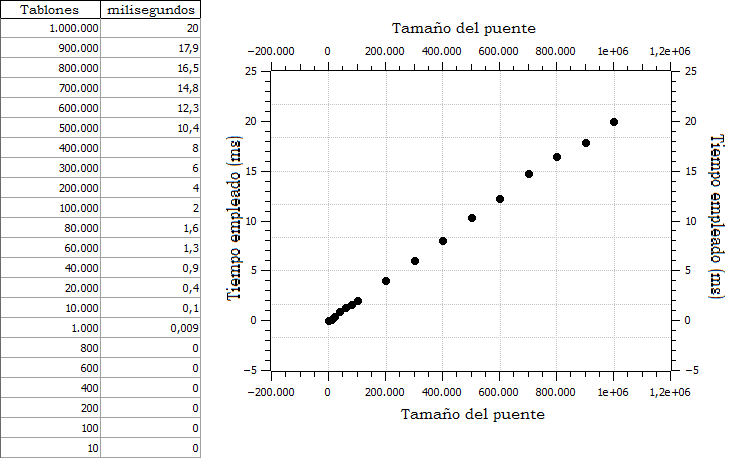
Este caso, le toma al algoritmo recorrer todo el puente, por lo que, si n es la cantidad de tablones, el algoritmo es O(n).

Notar que en este caso tampoco tiene sentido variar la longitud del salto ya que si no, no estaríamos evaluando el peor caso.

A diferencia del mejor caso, aquí tampoco podemos variar el estado de los tablones, ya que estos deben ser todos válidos, para que, dando saltos de 1 de longitud, el participante pueda cruzar todo el puente, forzando al algoritmo a su peor caso.

Por ende, el único parámetro a variar para realizar pruebas, es la longitud del puente.

Luego de realizar las pruebas pertinentes de este escenario, (y de realizar 10.000 veces cada una) como declaramos al inicio del insiso, mostramos los resultados obtenidos en el siguiente gráfico:



En la figura superior se puede apreciar que el tiempo de resolución de peor caso del algoritmo es lineal al tamaño del puente de entrada.

Escenarios de casos promedio del algoritmo.

En estos escenarios, decidimos evaluar y tomar métricas para casos no tan “border” como los de los escenarios anteriores. La idea es tomar muestras del algoritmo haciendo variar la longitud de salto del participante como el tamaño del puente, generando de manera aleatoria el estado de los tablones del mismo.

Para diferenciar bien los casos y poder analizar mejor, decidimos que cada escenario de las pruebas de caso promedio tenga una capacidad de salto constante. En cada uno de estos escenarios veremos cómo responde el algoritmo a medida que varía el tamaño de la entrada.

Notar que a diferencia del mejor y del peor caso, aquí no está garantizado que el participante vaya a cruzar todo el puente. En los casos anteriores, cruzar el puente era requerido para poder contemplar el caso de análisis, tanto para el mejor como para el peor caso. Para poder ver en los gráficos, no solo el tiempo de respuesta del algoritmo en función de la entrada, sino también si el participante pudo o no cruzar el puente, se marcó de color verde a las instancias en las que sí pudo, y de color rojo a las que no.

Como era de esperar, como el estado de cada tablón es aleatorio, mientras más pequeño es el salto del participante, mayor es la probabilidad de que este no pueda cruzar el puente, ya que la probabilidad de que haya tantos tablones inválidos consecutivos como la capacidad de salto, aumenta.

Dicha probabilidad también aumenta cuando el puente crece, ya que hay un mayor espacio para que dicha secuencia de tablones inválidos aparezca.

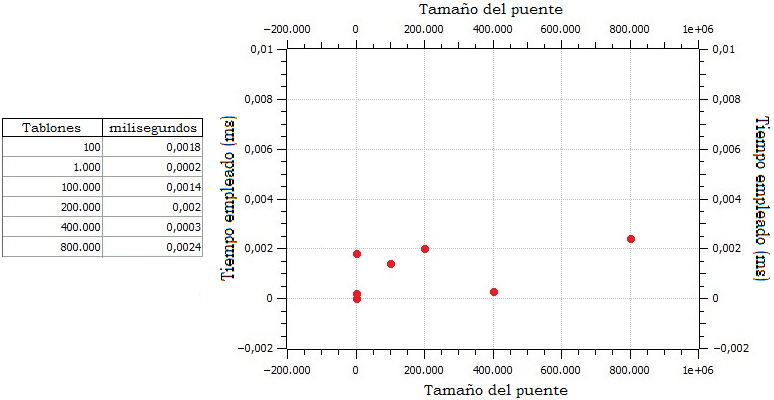
Como fue mencionado anteriormente, para realizar estas pruebas, se diseñó un generador de instancias de caso promedio, que dado un tamaño de entrada y una capacidad de salto, generaba tantas instancias distintas como se quisiera, respetando el tamaño asignado. Eso nos permitió evaluar la respuesta del algoritmo para distintas instancias del mismo tamaño, y poder tomar un promedio.

Para estos casos, también cada prueba de cada escenario fue repetida un total de 10.000 veces para aminorizar el margen de error producido debido a que el CPU no está atendiendo únicamente nuestro proceso.

Como última observación a hacer, es interesante notar que para los casos en los que el participante no pueda cruzar todo el puente, los tiempos no siguen ningún patrón ni concordancia, ya que, si llamamos C a la capacidad de salto de un participante, el hecho de que haya C tablones rotos consecutivos es totalmente aleatorio ya que el estado de los tablones fue generado de manera aleatoria.

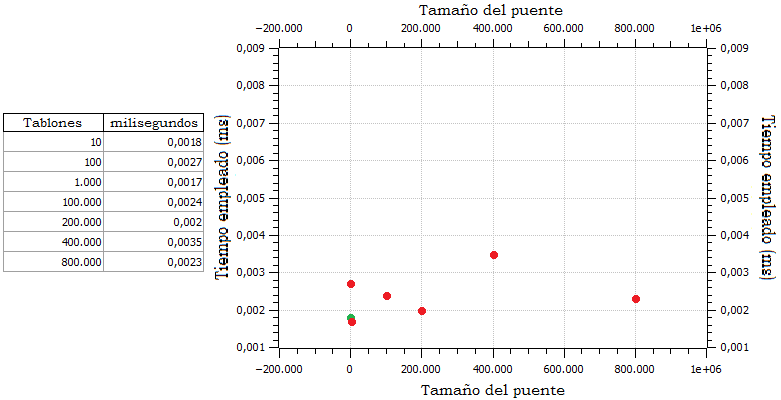
Sin más, presentamos los distintos escenarios.

ESCENARIO DE SALTO = 2



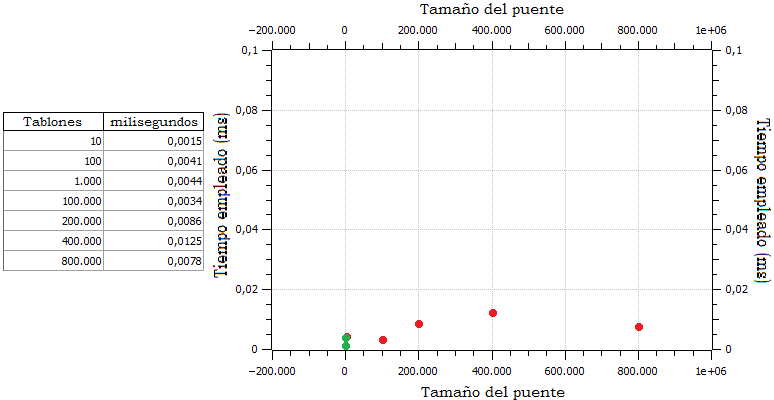
Tal como lo expresado anteriormente, para saltos de baja longitud, la probabilidad de no poder cruzar el puente aumenta, por lo que los tiempos del algoritmo pueden variar mucho de instancia a instancia.

ESCENARIO DE SALTO = 4



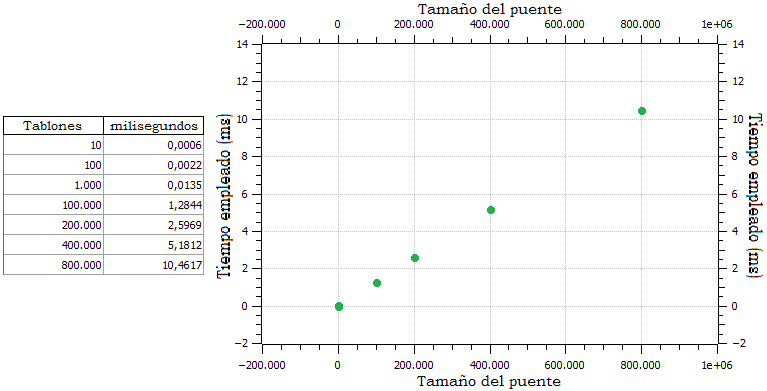
Aquí se puede apreciar que una de las instancias generadas permitió al participante cruzar todo el puente, y que fue la instancia en la que más probabilidades tenía de hacerlo, es decir, la instancia con un puente de longitud = 10. Para el resto, continuaron apareciendo de manera aleatoria distintos baches en el puente que no permitieron al jugador terminar de cruzar el mismo. Notar que los baches se hicieron presentes al poco tiempo de recorrer el puente, tal como era esperado probabilísticamente.

ESCENARIO DE SALTO = 8



En saltos de tamaño 8, se continuó observando que la probabilidad de poder cruzar el puente en altas instancias continúa siendo muy baja y el algoritmo termina rápidamente debido a eso. Aunque como cada instancia no depende de las demás, no podemos acotar al tiempo del algoritmo, pero si podemos hacerlo, justificando que cierta cota es válida para el caso promedio, para saltos <= 8.

ESCENARIO DE SALTO = 16

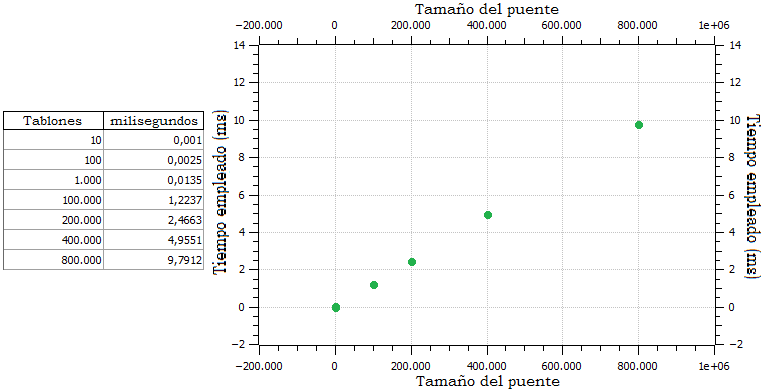


A partir de aquí, notamos que con las mismas longitudes de prueba que en los escenarios anteriores, no se produjo una caída del puente, en ninguna de las distintas instancias que se probaron para cada uno de los tamaños (10, 100,…,800.000). Vale destacar, que con un salto de longitud = 16, se cae en el mejor caso, para la instancia de 10 tablones, ya que puede cruzarla toda de una, haciendo que la probabilidad de caer del puente pase a ser nula. Lo mismo ocurrirá con luego con saltos mayores, al superar a la instancia de 100 tablones.

Si bien en todas las pruebas de este escenario el participante pudo cruzar, eso no escapa de la probabilidad de que la generación aleatoria pueda dar 16 tablones rotos consecutivos. Por lo que no podemos dar una cota fija, pero podríamos estimar que en el caso promedio, el puente puede ser cruzado, lo que da una complejidad de O(n) para el caso promedio, con n igual a la cantidad de tablones.

Es O(n) ya que el algoritmo debe recorrer todo el puente para terminar.

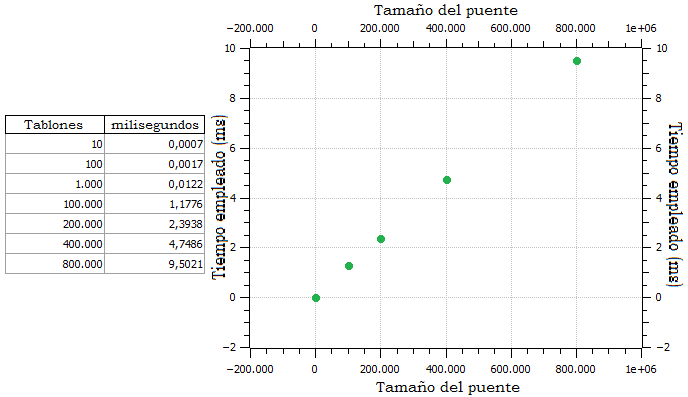
ESCENARIO DE SALTO = 32



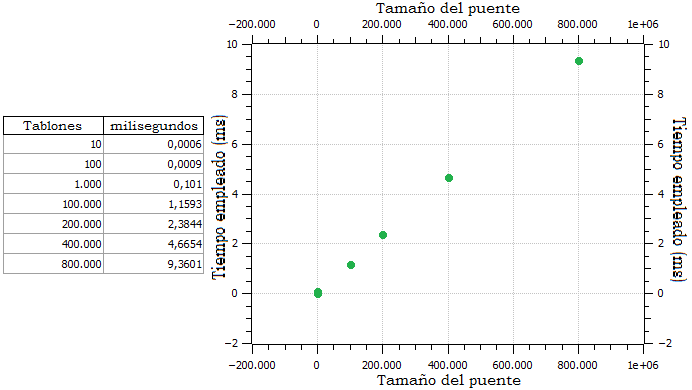
Aquí y en adelante, la probabilidad de que haya una cantidad de tablones rotos consecutivos igual a la cantidad del salto, es realmente baja, más allá del tamaño de la entrada; tendría que ser una entrada exageradamente grande para que la probabilidad aumente apenas un poco. Por ende de aquí en adelante, afirmamos que el caso promedio del algoritmo es O(n), (recordando que n es la cantidad de tablones del puente en cuestión).

Por lo dicho recién, en los siguientes escenarios era esperado que la respuesta del algoritmo sea líneal al tamaño de la entrada.

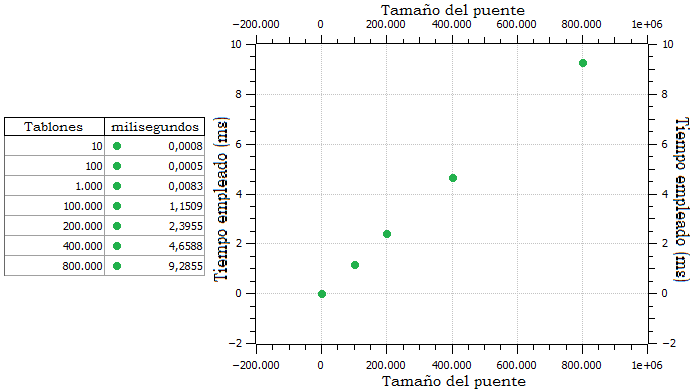
ESCENARIO DE SALTO = 64



ESCENARIO DE SALTO = 128



ESCENARIO DE SALTO = 256



ALGUNAS CONCLUSIONES

Haciendo un análisis probabilístico de nuestro algoritmo, por lo antes explicado podemos afirmar que en saltos grandes, el tiempo de computo de una instancia es lineal al tamaño del puente de la misma.

En cuanto a saltos pequeños no hay un patrón que se mantenga ya que el “hueco” sin tablones válidos por los cuales no se podría cruzar es totalmente al azar, pero probabilísticamente, si el salto es chico, dicho “hueco” se hará presente al poco tiempo de computo, por lo que el algoritmo responde eficientemente y podríamos acotar este tiempo para un caso promedio.