Universidad Nacional Autónoma de México

Análisis de Algoritmos
Semanal 2

Luis Sebastián Arrieta Mancera (318174116)
Zuriel Enrique Martínez Hernández (318056423)

5 de febrero de 2023

Ejercicio

En la nueva edición de El juego del calamar, un juego consiste en formar grupos de 100 personas. Cada grupo tendrá su turno para jugar. El objetivo del juego es que en cada grupo los jugadores se eliminen entre ellos hasta quedar solo uno vivo.

Las reglas del juego, vigiladas muy cerca por los guardias para que nadie las rompa, son las siguientes:

1. Las 100 personas deben formar un círculo. A alguna persona se le asignará el número 1 y, a partir de ella, se le asignará el número $i + 1$ a la persona a la izquierda de la que tiene el número i hasta haber enumerado a las 100 personas.
   Queda prohibido durante el juego abandonar su posición en el círculo.

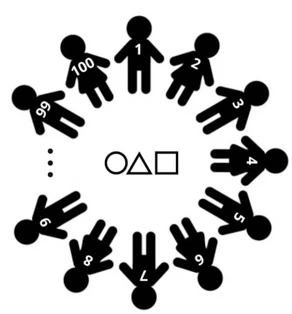

Figura 1: Círculo de personas.

2. Se le informa a los jugadores que si reciben un arma:
   a) Deben eliminar a la persona viva más cercana a su izquierda.
   b) Después de haberla eliminado, debe darle el arma a la persona viva más cercana a su izquierda.

Preguntas

1. Si al iniciar el juego, le dan un arma a la persona etiquetada con el **número 1** (y solo a esa persona):

   - Menciona quién sera el sobreviviente. Justifica tu respuesta.

     El sobreviviente será el número 73. Haciéndolo manualmente, tenemos que:

```
INICIO:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85
86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

PRIMER RONDA:
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37
39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69
71 73 75 77 79 81 83 85 87 89 91 93 95 97 99

SEGUNDA RONDA:
1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69
73 77 81 85 89 93 97

TERCER RONDA:
9 17 25 33 41 49 57 65 73 81 89 97

CUARTA RONDA:
9 25 41 57 73 89

QUINTA RONDA:
9 41 73

FIN
73
```

     Podemos notar que conforme pasan las rondas el numero de jugadores se divide a la mitad. En la primer ronda pasan de ser 100 a ser 50, luego 25, luego 12, 6, 3 y por ultimo 1.

     También notemos que cuando el numero de jugadores es par el primer jugador siempre sobrevivirá, es por eso que el número 1 sobrevive hasta la segunda ronda, pues cuando el numero de jugadores es impar, el arma llegará al ultimo jugador en la lista y como ya no hay otro jugador al lado, tiene que eliminar al primero en la lista, en este caso el numero 1.

     De este modo, si recapitulamos a partir de la cuarta ronda, nos damos cuenta que hay un numero par de jugadores, entonces como el jugador 9 empieza con el arma, al final de la ronda el arma volverá a él, pues el 73 eliminó al 89.

     Finalmente, en la quinta ronda ahora tenemos un número impar de jugadores, por lo que el 9 eliminaría al 41 y le pasa el arma al 73, pero como ya no quedan más jugadores, el 73 tiene que eliminar al 9 y así, se convierte en el sobreviviente.

- Si el juego consistiera en n personas, brinda un algoritmo que indique quién será el sobreviviente en tiempo $\Theta(n)$.

  El juego finaliza cuando solo haya una persona viva. Entonces se harán n-1 asesinatos. Podemos tener un ciclo *for* desde 0 hasta n-1 y por cada iteración eliminar a una persona. Como es un círculo, al eliminar a una persona se retira el cadáver y el círculo se cierra. Podemos representar este comportamiento con una lista. Solo hay que considerar los casos en los que el índice excede el limite n-1.

Algorithm 1 Algoritmo para encontrar al ganador en $T(n) = \theta(n)$

```
Require: : jugadores = [j_1, j_2, ..., j_n]
Require: : inicial = indice-jugador-inicial

n ← length(jugadores)
p ← inicial                                   ▷ posicion del arma
v ← n                                         ▷ cantiad de jugadores vivos
for i = 0 to n − 1 do
    if (p + 1) == v then p = −1               ▷ si el indice del jugador a matar excede v
    end if
    del jugadores[p + 1]                      ▷ eliminamos al jugador de la izquierda
    v− = 1
    p+ = 1
    if p == v then p = v mod p                ▷ si la posicion del arma excede v
    end if
end for

return jugadores[0]
```

- Demuestra que si el círculo fuera de 2^k personas, con $k \in \mathbb{N}$, el sobreviviente será el jugador etiquetado con el número 1.

  Sabemos que por cada vuelta que de el arma, el número de jugadores se reducirá a la mitad. Suponiendo que $n = 2k$ (par) los jugadores que tendrán el arma se representa en la secuencia $\{1,(1+2),(1+2+2),(1+2+2+2),...,2k-1\}$, visto de otra forma $\{1, 3, 5, 7, ..., 2k-1\}$ puesto que los jugadores en tener el arma serán los etiquetados con números de la forma $2k+1$ y el jugador $2k$ será asesinado a manos del jugador $2k-1$ y el siguiente jugador en tener el arma será el jugador 1, de esta manera el jugador 1 sobrevive a la primera ronda. Ahora bien, suponiendo que $n = 2k+1$ (impar) y por lo anterior, el último jugador en tener el arma es el jugador en la posición $2k+1$, por lo que el siguiente jugador en morir es el jugador en la posición 1. De esta manera sabemos que cuando la cantidad de jugadores es par el jugador 1 sobrevive, y cuando es impar muere.

  Por lo anterior, si queremos que el jugador 1 sobreviva, necesitamos un número que tenga la particularidad de que sin importar cuantas veces se divida entre dos el resultado sea un número par ¿Qué números podrán cumplir con esta propiedad?. En efecto, los número que son de la forma 2^k, es decir, las potencias de dos puesto que si $n = 2^k = 2*2^{k-1} = 2*2*2^{k-2} = 2*2*...*2^{k-k} = 2*2*...*2^0 = 2*2*...*1 = 2*(2*...*1)$ donde $a = (2*...*1)$, como $n = 2a$ y por hipótesis a también es par, entonces si n es potencia de dos, entonces el jugador en iniciar el juego sobrevivirá. En particular para este ejemplo el jugador 1 sobrevivirá.

Demostracion por induccion.

Caso Base:

Sea $k = 1$ entonces $2^1 = 2$, por las reglas del juego, el jugador 1 recibiría el arma y mataría al jugador 2 quedando así como el sobreviviente.

Hipótesis Inductiva:

Suponemos que se cumple para 2^k.

Paso Inductivo:

Tenemos que 2^{k+1}, pero sabemos que esto lo podemos ver como $2^k \cdot 2$. Ahora, por **H.I** sabemos que 2^k se cumple, lo que quiere decir que el jugador 1 sobrevivió. Entonces, tenemos que $1 \cdot 2 = 2$ pero este es nuestro caso base, por lo que se cumple que el jugador 1 es el sobreviviente.

2. Si al iniciar el juego, le dan un arma a la persona etiquetada con el **número r** (y solo a esa persona):

   - Si el juego consistiera en n personas, brinda un algoritmo que indique quién será el sobreviviente en tiempo $o(n)$.

     Por la demostración del inciso anterior sabemos que si la cantidad de jugadores es una potencia de dos, entonces el jugador inicial ganará el juego. Cuando la cantidad de jugadores no cumple con esta condición, tenemos que determinar qué jugador inicia la ronda cuando la cantidad de jugadores se reduce a una potencia de dos.

     Para encontrar la potencia de dos más cercana tenemos que calcular el logaritmo en base dos de la cantidad de jugadores ($log_2 n$). Redondeamos el resultado al piso, esto cubre el caso en el que n no sea una potencia de dos. Usamos el resultado como exponente de la base dos y determinamos el número de muertes necesarias para que la cantidad de jugadores vivos sea de la forma 2^k, esto se hace restando n menos la potencia de dos que calculamos previamente. Como sabemos que cada 2 lugares un jugador es asesinado entonces el ganador será la suma del número del jugador inicial más dos veces el número de muertes que calculamos, también hay que considerar hacer una conversión cuando la posición ganadora exceda el número de jugadores (n), esto para que el desborde caiga en una posición valida dentro de nuestro conjunto de jugadores.

Algorithm 2 Algoritmo para encontrar al ganador en $T(n) = o(n)$

Require: n ▷ cantidad de jugadores
Require: r ▷ posicion incial del arma

exp ← floor(log(n,2)) ▷ $log_2(n)$
powerOfTwo ← pow(2,exp)
deaths ← (n−powerOfTwo)
ganador ← ((deaths * 2) + r) **mod** n ▷ **mod** para evitar index out of bound

return ganador

Notemos que las operaciones que realizamos en este algoritmo son **çonstantes"** y no hay ciclos for, por lo que la complejidad del algoritmo es en tiempo o(n)

Nota: En el código para esta actividad se usa una lista de jugadores como entrada del método y se calcula n, mientras que en este pseudocódigo se considera que solo te dan el número de jugadores, esto para optimizar el algoritmo, en el código igual se puede implementar de esta manera pero decidimos dejarlo así.

- Si el juego consistiera en n personas, tú eres parte del juego, a ti te tocó ser el número p en el círculo y, por diversión, los guardias te dan la oportunidad de decir a qué jugador deben entregarle el arma para comenzar el juego, ¿qué número r deberías decirles si quieres sobrevivir?

  Gracias al inciso anterior ya sabemos la fórmula para encontrar al ganador. En esencia esto consiste en calcular cuantos jugadores deben de morir hasta que la cantidad de jugadores sea potencia de dos. Entonces si somos la posición p, tenemos que decirle a los guardias que le den el arma al jugador que está k posiciones antes de nosotros, es decir k jugadores a la derecha. Donde k es el doble de jugadores que deben deben de morir hasta que la n sea potencia de dos. El algoritmo para calcular esto es el siguiente

Algorithm 3 Algoritmo para salvar nuestra vida en $T(n) = o(n)$

Require: n ▷ cantidad de jugadores
Require: p ▷ nuestra posicion

exp ← floor(log(n,2)) ▷ $log_2(n)$
powerOfTwo ← pow(2,exp)
deaths ← (n−powerOfTwo)
places ← deaths $*$ 2 ▷ lugares a deplazar
r ← p - places ▷ persona a la que le tenemos que dar el arma

if r $<$ 0 **then** r $= n + p$ ▷ si la posicion es negativa
end if

return r