

**UNVIME Escuela de Ingeniería y Ciencias Ambientales - Ingeniería en Sistemas**

**Materia: Análisis y Diseño de Algoritmos**

**Alumno: Gimenez Thomas Valentin**

**Desarrollo:**

**Ejercicio 1**

Si lo analizamos de manera lógica, el bote solo puede contener o un soldado o los dos niños al mismo tiempo

Suponiendo que los niños podemos llamarlos x1 y x2 y a los soldados como n, la solución para que los soldados crucen seria la siguiente secuencia

1. Cruzan por la orilla el niño x1 con un soldado n1 desde la orilla a
2. El soldado n se baja en la orilla b y el niño x1 vuelve solo a la orilla b
3. Se sube un soldado n2 al bote con el niño x1 desde la orilla a

\*Esto se repite hasta que llegue el soldado n\*

1. El niño x1 se baja en la orilla b y el soldado n vuelve a la orilla a
2. El niño x2 se sube con el soldado n y cruzan hasta llegar a la orilla b

Entonces, un soldado necesita 2 cruces ya que es 1er cruce donde llega el soldado a la orilla b y el 2do cruce para cuando el niño vuelve a la orilla a

Por lo que el total de cruces son 2n sumando los 2 cruces para que el niño x2 pueda volver a la orilla b

Osea que serian **2n+2** cruces

**Ejercicio 2**

Nosotros sabemos que los algoritmos por inserción directa (insertion sort) es un algoritmo que nos permite realizar el ordenamiento de tal manera que podamos resolver el ejercicio que se nos pide. En este caso, la cantidad de comparaciones depende de la entrada que se nos da (2n vasos ordenados de manera alternada)

Por lo que visto algorítmicamente (V para vasos llenos y E para vasos vacíos)

Dado que el Vaso V1 que esta lleno, entonces esta “ordenado”

Entonces las siguientes iteraciones van a ser tal que:

Si el elemento del bucle i es V, y al lado tiene un elemento E, entonces traslada V a la derecha de V

En el caso de que en i sea E, entonces se deja en su lugar

El número de movimientos totales son:

Primer vaso lleno no se mueve

El segundo vaso lleno se mueve un paso hacia la derecha

El tercer vaso lleno se mueve 2 pasos hacia la derecha

Y así hasta que se mueva n-1 veces

Por lo que los movimientos totales van a ser **(n-1)**

Y el número de comparaciones clave para este tipo de entrada va a ser:

b. Si los vasos están en orden aleatorio, el algoritmo seria el mismo (insertion sort), por lo que la cantidad de movimientos dependerá del orden inicial en el que se encuentren los vasos en cuestión. Si lo vemos así, analizaríamos el peor de los casos, ósea, cuando se requieran la mayor cantidad de movimientos

Por lo que la cantidad de movimientos va a ser de complejidad 0

c.

i. El centinela puede ser A o todo valor que sea menor que A (cualquier valor numérico o que sea menor a A en cuanto a valor ASCII) ya que se necesita ser insertado al inicio del array

ii. Ambos tienen la complejidad parecía 0 **,** pero en cuanto a rendimiento es mejor la versión con centinela ya que obvia un condicional (j>0) en las iteraciones del bucle interno