

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA
3ra. práctica (tipo B)
(Segundo Semestre 2021)

Indicaciones Generales:

- Duración: 2h 50 min.
- Al inicio de cada programa, el alumno deberá incluir, a modo de comentario, la estrategia que utilizará para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- Los programas deben ser desarrollados en Ansi C. Si la implementación es diferente a la estrategia indicada o no la incluye, la pregunta no será corregida.
- Un programa que no muestre resultados coherentes y/o útiles será corregido sobre el 50% del puntaje asignado a dicha pregunta.
- Debe utilizar comentarios para explicar la lógica seguida en el programa elaborado.
- El orden será parte de la evaluación.
- Se utilizarán herramientas para la detección de plagios, por tal motivo si se encuentran soluciones similares, se anulará la evaluación a todos los implicados y se procederá con las medidas disciplinarias dispuestas por la FCI.
- Para este laboratorio solo se permite el uso de las librerías **stdio.h**, **stdlib.h** y **math.h**
- Su trabajo deberá ser subido a PAIDEIA en el espacio indicado por los jefes de práctica.
- Los archivos deben llevar como nombre su código de la siguiente forma **codigo_LabX_PY**

Pregunta 1 (10 puntos)

Se desea implementar un programa que gestione la asignación de traslaciones de dos ascensores de un mismo edificio de N pisos. Para ello, el programa utiliza una cola general de solicitudes en el que cada solicitud tiene el piso al que debe dirigirse y su dirección si es de subida o de bajada. También se cuenta con dos colas de solicitudes adicionales, una para cada ascensor. El programa sólo se encargará de asignar las solicitudes desde la cola general a las colas individuales de cada ascensor. Se tiene como consideración para este edificio que uno de los ascensores sólo recibirá solicitudes de subida y el otro ascensor sólo recibirá solicitudes de bajada. Luego que el ascensor llega a un piso pueden haber más solicitudes generales para cualquiera de los dos ascensores debido a las llamadas desde otros pisos. Recuerde que cuando una persona está fuera de los ascensores en un piso y desea solicitar al ascensor, puede presionar uno de los dos botones si es para subir o bajar, e incluso puede darse el caso de haber dos solicitudes desde un piso, sea para subir y bajar, o bajar y subir, pero una ocurre después de la otra. Si una persona presiona más una vez el mismo botón, solo se considera una solicitud y es encolada. Para el programa, no considere las solicitudes que una persona pueda realizar cuando ya está dentro del ascensor.

Para que la gestión de las traslaciones de los ascensores se realice, luego que se ingresan las solicitudes a la cola general, se desencola una por una, si una solicitud es de subida esta se añade a la cola del ascensor que solo recibe solicitudes de subida, en el que luego las solicitudes deben de ordenarse de menor a mayor aquellas que son superiores al piso donde se encuentra el ascensor hasta el máximo piso ingresado en la cola comenzando por la cabeza de la cola si es que la solicitud que está en la cabeza va de subida. Cuando se extrae una solicitud de la cola general y esta es de bajada, se añade a la cola del otro ascensor que sólo recibe solicitudes de bajada. Las solicitudes de bajada que son inferiores al piso donde se encuentra el ascensor se ordenan de mayor a menor hasta el mínimo piso ingresado hasta ese momento, y luego siguen de mayor a menor las solicitudes que están por encima del piso actual.

Su programa realizará la simulación de las solicitudes y movimientos del ascensor mientras existan solicitudes en la cola general. Asuma que al inicio ambos ascensores están en el primer piso y tiene en

la cola una solicitud ingresada que va de subida hacia el primer piso. Para la simulación, cuando cualquiera de los ascensores está en un piso, debe de indicar la cantidad de solicitudes a ingresar a la cola general, por cada solicitud se indica el piso a dirigirse y el valor de la dirección (1 si es de subida o 0 si es de bajada).

A continuación, les mostramos un ejemplo del ingreso de datos que requiere este programa:

Ingrese la cantidad de pisos del edificio: 7
Movimientos del ascensor en un edificio de 7 pisos:

La cola general de solicitudes tiene:
1-1 ->
Luego de asignar las solicitudes a cada ascensor se tiene:
Cola 1: 1-1 ->
Cola 2: 1-1 ->
El ascensor 1 está en el piso 1
El ascensor 2 está en el piso 1
Ingrese la cantidad de solicitudes: 5

Solicitud 1 [número_piso y dirección(1:sube,0:baja)]: 3 1
Solicitud 2 [número_piso y dirección(1:sube,0:baja)]: 6 1
Solicitud 3 [número_piso y dirección(1:sube,0:baja)]: 2 1
Solicitud 4 [número_piso y dirección(1:sube,0:baja)]: 4 1
Solicitud 5 [número_piso y dirección(1:sube,0:baja)]: 7 0

La cola general de solicitudes tiene:
7-0 -> 4-1 -> 2-1 -> 6-1 -> 3-1 ->
Luego de asignar las solicitudes a cada ascensor se tiene:
Cola 1: 6-1 -> 4-1 -> 3-1 -> 2-1 ->
Cola 2: 7-0 ->
El ascensor 1 está en el piso 2
El ascensor 2 está en el piso 7

Ingrese la cantidad de solicitudes: 4
Solicitud 1 [número_piso y dirección(1:sube,0:baja)]: 6 0
Solicitud 2 [número_piso y dirección(1:sube,0:baja)]: 5 1
Solicitud 3 [número_piso y dirección(1:sube,0:baja)]: 1 1
Solicitud 4 [número_piso y dirección(1:sube,0:baja)]: 3 0

La cola general de solicitudes tiene:
3-0 -> 1-1 -> 5-1 -> 6-0 ->
Luego de asignar las solicitudes a cada ascensor se tiene:
Cola 1: 1-1 -> 6-1 -> 5-1 -> 4-1 -> 3-1
Cola 2: 3-0 -> 6-0
El ascensor 1 está en el piso 3
El ascensor 2 está en el piso 6

Ingrese la cantidad de solicitudes: 3
Solicitud 1 [número_piso y dirección(1:sube,0:baja)]: 7 0
Solicitud 2 [número_piso y dirección(1:sube,0:baja)]: 2 1
Solicitud 3 [número_piso y dirección(1:sube,0:baja)]: 4 0

La cola general de solicitudes tiene:
4-0 -> 2-1 -> 7-0 ->
Luego de asignar las solicitudes a cada ascensor se tiene:
Cola 1: 2-1 -> 1-1 -> 6-1 -> 5-1 -> 4-1
Cola 2: 7-0 -> 3-0 -> 4-0
El ascensor 1 está en el piso 4
El ascensor 2 está en el piso 4

Ingrese la cantidad de solicitudes: 3
 Solicitud 1 [número_piso y dirección(1:sube,0:baja)]: 5 0
 Solicitud 2 [número_piso y dirección(1:sube,0:baja)]: 6 0
 Solicitud 3 [número_piso y dirección(1:sube,0:baja)]: 3 1

 La cola general de solicitudes tiene:
 3-1 -> 6-0 -> 5-0 ->
 Luego de asignar las solicitudes a cada ascensor se tiene:
 Cola 1: 3-1 -> 2-1 -> 1-1 -> 6-1 -> 5-1
 Cola 2: 5-0 -> 6-0 -> 7-0 -> 3-0
 El ascensor 1 está en el piso 5
 El ascensor 2 está en el piso 3

 Ingrese la cantidad de solicitudes: 3
 Solicitud 1 [número_piso y dirección(1:sube,0:baja)]: 2 0
 Solicitud 2 [número_piso y dirección(1:sube,0:baja)]: 7 1
 Solicitud 3 [número_piso y dirección(1:sube,0:baja)]: 4 1

 La cola general de solicitudes tiene:
 4-1 -> 7-1 -> 2-0 ->
 Luego de asignar las solicitudes a cada ascensor se tiene:
 Cola 1: 7-1 -> 4-1 -> 3-1 -> 2-1 -> 1-1 -> 6-1
 Cola 2: 2-0 -> 5-0 -> 6-0 -> 7-0
 El ascensor 1 está en el piso 6
 El ascensor 2 está en el piso 7

 Ingrese la cantidad de solicitudes: 0

 Luego de asignar las solicitudes a cada ascensor se tiene:
 Cola 1: 7-1 -> 4-1 -> 3-1 -> 2-1 -> 1-1 ->
 Cola 2: 2-0 -> 5-0 -> 6-0 ->
 El ascensor 1 está en el piso 1
 El ascensor 2 está en el piso 6

 Ingrese la cantidad de solicitudes: 0

 Luego de asignar las solicitudes a cada ascensor se tiene:
 Cola 1: 7-1 -> 4-1 -> 3-1 -> 2-1 ->
 Cola 2: 2-0 -> 5-0 ->
 El ascensor 1 está en el piso 2
 El ascensor 2 está en el piso 5

 Ingrese la cantidad de solicitudes: 0

 Luego de asignar las solicitudes a cada ascensor se tiene:
 Cola 1: 7-1 -> 4-1 -> 3-1 ->
 Cola 2: 2-0 ->
 El ascensor 1 está en el piso 3
 El ascensor 2 está en el piso 2

 Ingrese la cantidad de solicitudes: 0

 Luego de asignar las solicitudes a cada ascensor se tiene:
 Cola 1: 7-1 -> 4-1 ->
 Cola 2:
 El ascensor 1 está en el piso 4
 El ascensor 2 está en el piso 2

 Ingrese la cantidad de solicitudes: 0

 Luego de asignar las solicitudes a cada ascensor se tiene:

Cola 1: 7-1 ->

Cola 2:

El ascensor 1 está en el piso 7

El ascensor 2 está en el piso 2

Ingrese la cantidad de solicitudes: 0

Luego de asignar las solicitudes a cada ascensor se tiene:

Cola 1:

Cola 2:

El ascensor 1 está en el piso 7

El ascensor 2 está en el piso 2

Se le solicita implementar lo siguiente:

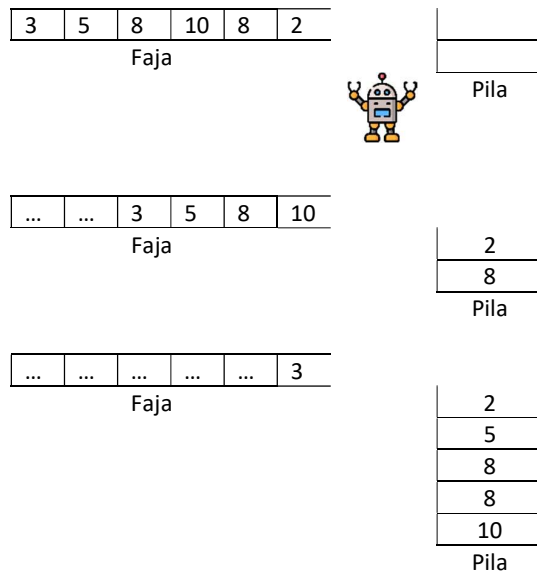
- a. Las estructuras de datos que permitan soportar lo descrito (2.0 puntos).
- b. Una función que permita ordenar la cola de acuerdo con las consideraciones descritas. Para el ordenamiento de la cola sólo puede usar las operaciones de cola (encolar, desencolar, está vacía y ojear la cabeza). No puede usar otras estructuras como arreglos, listas, pilas u otras colas para resolver el ordenamiento de la cola, solo la misma cola de solicitudes. Puede usar iteraciones o recursividad (5.5 puntos).
- c. Las funciones necesarias incluyendo el principal para realizar la simulación descrita (2.5 puntos).

Pregunta 2 (10 puntos)

Una empresa adquiere un nuevo robot apilador modelo Hanoi 2.0, esta unidad se encargará de mover los productos que vienen por una faja transportadora (opera como cola) hacia una pila donde debe acumular todos los productos, para tal tarea el robot tiene 2 brazos mecanizados, uno de ellos se encargará de recibir el producto que ha salido de la cola y el otro servirá para realizar los movimientos necesarios para mantener la pila ordenada. La pila siempre estará ordenada colocando los productos con mayor peso en la parte inferior y los productos de menor peso en la parte más alta ya que, debido a su fragilidad, si se coloca un producto con menor peso bajo un producto de mayor peso, el mas ligero se puede quebrar.

Aunque esta tarea puede ser difícil para cualquier robot, nuestra nueva unidad Hanoi 2.0, cuenta con muchas herramientas avanzadas, como es una memoria que le permite registrar todos los productos que va apilando, la cual se puede manejar como a un arreglo y la posibilidad de utilizar estrategias de recursión y de iteración sin problema, para cualquier tarea que lo necesite. Además, tiene incorporados algoritmos de ordenamiento como bubble sort, quick sort y merge sort.

Para realizar la tarea de mantener siempre la pila ordenada por peso, el robot puede usar 2 pilas auxiliares, donde puede colocar los productos temporalmente antes de llevarlos a la pila principal. Desde luego la restricción de fragilidad de los productos debe ser considerado tanto para la pila principal, como para sus 2 auxiliares, por lo que en ningún momento puede haber un producto de mayor peso sobre un producto de menor peso. A continuación, mostramos un ejemplo de una línea y como los productos se van acumulando en la pila:



Los valores en cada nodo de la cola y la pila estarán dados solo por el peso. Recuerde que se tratan de objetos reales por tal motivo **no** es posible registrar en la memoria del robot los valores de los productos y luego colocarlos en la pila, pero si puede utilizar este espacio para realizar tareas de registro de los productos apilados o verificar los pesos existentes en la pila, ya que esta última no se puede recorrer.

- Desarrolle las sentencias y estructuras necesarias para el ingreso de datos en la faja transportadora. (2 puntos)
- Implemente un programa para el robot apilador, que sirva para colocar los elementos de la faja transportadora en la pila principal, considerando siempre la restricción de fragilidad al realizar los movimientos. Para esta tarea solo puede usar 2 pilas auxiliares y variables de tipo entero. Desde luego también puede usar la memoria de la unidad, pero solo como registro de datos (8 puntos).

Profesor del curso: Johan Baldeón
 Rony Cueva

San Miguel, 29 de octubre del 2021