Reorganización de Megathlon.
Práctica otoño 2024

November 24, 2024

La empresa Megathlon desea cambiar la organización de sus tiendas probando nuevas distribuciones de sus productos, y formas de pago. Nos ha encargado que implementemos una simulación de un nuevo tipo de tienda para poder estudiar si adoptan este nuevo diseño en el futuro. Esta empresa trabaja con un gran número de productos diferentes, pero los clasifica en tipos de productos como piscina, alimentación deportiva, material de tenis, etc. Por ejemplo, el tipo de producto piscina, incluye bañadores de hombre, mujer y niño, gorros, chancletas, etc. De todas formas, en esta simulación, solamente trabajaremos con los tipos de productos, ignorando los productos que incluyen.

El nuevo tipo de tienda tendrá n salas diferentes identificadas por los nombres de los tipos de productos de la tienda. Solo se puede entrar por una única sala, la sala inicial o de entrada, cuyo identificador puede ser cualquiera. Desde cualquier sala se puede avanzar directamente o bien a dos salas, una por la izquierda y otra por la derecha, o bien a ninguna sala. Desde cualquier sala, menos la inicial, se puede retroceder directamente hasta una sala anterior. Cada sala tendrá un tipo de producto.

Al entrar en la tienda, el cliente encontrará una máquina. Allí apretará unos iconos que tienen los nombres de los tipos de productos de la tienda. A continuación se le expedirá un tiquet con un número identificativo de cliente (entero mayor o igual a 1), y un subárbol mínimo donde aparecerán márcadas todas las salas donde encontrar los tipos de productos que busca. Este subárbol mínimo siempre contendrá la raiz o sala inicial. Además, el tiquet le indicará como ir a la sala o salas para comprar los productos que necesita. Si necesita uno o más productos de entre los que se ofrecen, se le darán las indicaciones para llegar a las salas donde se encuentran los productos en forma de una secuencia de direcciones left/right/back con los nombres identificativos de las salas por donde se ha de pasar. Por tanto se ofrece un recorrido en profundidad del subárbol de salas, para que el cliente encuentre todos los productos que quiere comprar, empezando siempre por la sala inicial o de entrada.

La empresa está interesada en obtener para cada cliente el subárbol menor que contiene todas las salas donde se encuentran todos los productos que necesita el usuario, incluyendo la sala de entrada. La empresa quiere en un futuro guardar esta información para ir mejorando el diseño, de tal manera que para los

productos que se compran normalmente juntos, el subárbol sea lo más pequeño o lo más grande posible. Si los subárboles tienden a ser pequeños, los clientes serán más eficientes, la experiencia mejor,y el cliente tendrá más incentivo para volver. Por otro lado, si los subárboles tienden a ser grandes, los clientes visitarán muchas salas, y puede que compren otros productos que en principio no pensaban comprar. Posteriormente se analizarán estos datos para modificar la ubicación de los tipos de productos, y ver que ubicaciones revierten en unos mayores beneficios.

Al final, los clientes pasarán por la zona de cajas para pagar. Allí se introducirá el identificador del cliente, y se leerán los tipos de productos que compra, y su cantidad. Dado que los clientes han pasado por diversas salas para encontrar los productos que quieren comprar, puede que compren productos que no pensaban comprar. Por tanto, para cada cliente, se dará información sobre si han comprado productos que no tenía intención de comprar.

Otra mejora que Megathlon quiere implementar es organitzar las cajas de manera que se minimice el tiempo que transcurre desde que un cliente se acerca a la zona de cajas hasta que sale del establecimiento. En el centro de la zona de cajas, el usuario introduce su número de cliente y recoge un tiquet que tiene la hora de recogida del tiquet, la hora de salida de la tienda, y la caja donde ha de ir. El sistema nunca imprime más de un tiquet con el mismo instante. Además, en esta zona se leen los datos de los tipos de productos y cantidades que tiene el cliente en el carrito.

Las cajas se identifican con números entre 1 i el número de cajas. Al principio se lee el número de cajas que se van a utilizar en la simulación. La estrategia de asignación de cajas consiste en asignar el cliente a la caja abierta en la que tendrá que esperar menos. Si hay más de una con un mínimo tiempo de espera, se asignará a la caja con menos clientes asignados a ella (aunque todavía no hayan llegado). Si la primera y la segunda condición la cumplen más de una caja, se escogerá la que tenga un identificador más pequeño.

El tiempo que necesita un cliente para pagar es:

- El tiempo de desplazamiento desde el punto de recogida del tiquet hasta la caja asignada. Dado que los tiquets se cogen en el centro de la zona de cajas, el tiempo de ir a la caja que se le asigna es constante, y suponemos que es de cuatro segundos.

- El tiempo en caja, que se define como el número de productos que se va a comprar, más el tiempo de pago, que es aproximadamente unos 10 segundos en general. Asumimos que el tiempo de pasar por caja un producto siempre es el mismo para cada producto, es decir, 1 segundo.

- El tiempo que el cliente ha tenido que esperar en la cola, que es la suma de los tiempos de cobro de los clientes que estaban antes en la cola del cajero asignado, más lo que le queda al que está siendo ayudado. La caja puede no tener clientes en ese momento.

Respecto a los cálculos de tiempo, hemos de tener en cuenta lo siguiente:

- Si una caja ha atendido a un cliente hasta el segundo s, y hay un cliente en la cola, se empieza a atender al nuevo cliente en el segundo s+1.
- El tiempo de desplazamiento de un cliente comienza a contar a partir del instante marcado en el tiquet como hora de recogida del tiquet.
- Supondremos que la máquina de tiquets hace los cálculs inmediatamente, es decir, la cola se asigna tomando como referencia el estado del sistema a la hora que indica el tiquet.
- Supondremos que la simulación del paso por cajas ocurre en un día. Por tanto, no tenemos que gestionar horas repetidas (en días diferentes).

Al inicio se lee la estructura de salas con sus identificadores (tipos de productos). Después se lee el número de cajas. A partir que aquí se ejecutan las instrucciones que se detallan a continuación.

- nuevo_cliente: el dato de entrada es la lista de tipos de productos que el nuevo cliente quiere comprar. Los identificadores de cliente se otorgan de forma consecutiva a medida que entran los clientes en la tienda, a partir del 1. La salida de esta operación es el identificador del nuevo cliente, el subárbol mínimo que contiene las salas donde se encuentran los productos que el usuario quiere comprar (incluyendo la sala de entrada), y las instrucciones de cómo realizar el recorrido.

  Los datos de entrada serán correctos. Es decir, los identificadores de los tipos de productos que el cliente quiere comprar, pertenecerán a identificadores de tipos de productos leídos al leer la estructura de salas.
- quiere_comprar: el dato de entrada es un número de cliente. Se imprime la lista de tipos de productos que el cliente con el id leído quiere comprar, por orden alfabético. El número de cliente será correcto, es decir, será un entero mayor o igual a 1, que se habrá generado en una instrucción nuevo_cliente anterior.
- salas_a_visitar: el dato de entrada es un número de cliente. Se imprime la lista de identificadores de salas que el cliente con el id leído tiene que visitar para comprar los productos que necesita. Estos identificadores de sala se muestran por orden alfabético. El número de cliente será correcto, es decir, será un entero mayor o igual a 1, que se habrá generado en una instrucción nuevo_cliente anterior.
- escribir_caja: el dato de entrada es el número de caja y una hora. Se escribe la lista de los identificadores de clientes asignados a esa caja, con su hora de salida. Solo se escribirán los clientes que estén en caja o en la cola de esa caja a esa hora, o asignados a ella, eliminándose los clientes que ya hayan salido de la cola a esa hora. Los identificadores de cliente se escribirán en orden de llegada.

Los datos de entrada también serán correctos. El número de caja será un entero entre 1 y el número de cajas leído al inicio de la ejecución. La hora serán las horas, los minutos y los segundos separados por ":".

- escribir_cajas: el dato de entrada es la hora. Para cada caja, se escribe el identificador de la caja, y luego el contenido de ésta según se ha escrito en el item anterior. Las cajas se escriben por orden de identificador. En cada caja se eliminan los clientes que ya han sido atendidos a esa hora. Supondremos que la hora será correcta.

- pago_cliente: los datos de entrada son el número del cliente, la hora en la que se expende el tiquet, y la lista de tipos de productos y cantidades que se van a comprar. Primero se eliminan todos los clientes que ya han pasado por las cajas a la hora en que se expende el tiquet. Después se buscará la mejor cola para el cliente (según el criterio explicado con anterioridad), y se calculará la hora de salida. El output de esta operación es el número de cola donde ha de ir el cliente, y la hora de salida de la tienda.

  Los datos de entrada serán correctos. El número de cliente se habrá generado en una instrucción nuevo_cliente anterior y no será repetido, la hora en la que se expende el tiquet será correcta, los tipos de productos pertenecerán a identificadores de tipos de productos leídos al leer la estructura de salas, y las cantidades serán enteros mayores o iguales a 1. Para evitar ambigüedades con las horas, la simulación se hará en un solo día.

- compra_y_no_quiere: el dato de entrada es el identificador del cliente. La salida es el número total de items que el cliente ha comprado, de productos que no pensaba comprar. El número de cliente será correcto, es decir, se habrá generado en una instrucción nuevo_cliente anterior. Además, se habrá ejecutado la instrucción pago_cliente previamente para ese cliente, y por tanto, habrá información sobre lo que quería comprar, y lo que ha comprado.

- compran_y_no_quieren: no hay datos de entrada. La salida es la lista de identificadores de cliente, por orden de menor a mayor, que han comprado tipos de productos que no querían comprar.

- fin: marca el final de la ejecución.

Los tiempos de las instrucciones de escribir_caja, escribir_cajas y pago_cliente van en aumento. Es decir, si una instrucción se ejecuta antes que otra, el tiempo de la primera es anterior al tiempo de la siguiente. Los datos de entrada cumplirán esta condición.

Como ya hemos dicho anteriormente, a la hora de calcular a qué caja va un cliente, también consideraremos los clientes que han sido asignados a cajas, pero que aún no han llegado a su caja asignada. Lo mismo haremos para las opciones de escribir caja.