Ejemplos de uso del TAD cola

El problema de los palíndromos

Una frase es un **palíndromo** si se lee igual de izquierda a derecha que de derecha a izquierda, ignorando mayúsculas, espacios en blanco y signos de puntuación. El método que se va a presentar para averiguar si una frase introducida desde teclado y terminada por un punto es un palíndromo hace uso del TAD pila y del TAD cola.

Una pila es una estructura LIFO mientras que una cola es una estructura FIFO. Vamos a introducir las letras de la frase conforme son leidas en ambas estructuras. Al terminar, se extraerán letra por letra de cada estructura y si cada pareja de letras extraída coincide, la frase es un palíndromo.

```
procedimiento palíndromo
importa pilasDeCaracteres, colasDeCaracteres
variables p:pilaDeCar; c:colaDeCar

  procedimiento cargaFrase(sal p:pilaDeCar; sal c:colaDeCar);
  { Lee desde teclado una línea de caracteres y almacena las letras
    de la línea en p y en c por orden de introducción. }
  variable car: carácter
  principio
    creaPilaVacía(p); creaColaVacía(c);
    mientrasQue not finLínea hacer
      leer(car);
      si (car≥'a') and (car≤'z')
      entonces
        car:=chr(ord(car)+ord('A')-ord('a'))
      fsi;
      si (car≥'A') and (car≤'Z')
      entonces
        apilar(p,car);
        añadir(c,car)
      fsi
    fmq;
    nuevaLínea
  fin

  función chequeaPalíndromo(p:pilaDeCar; c:colaDeCar) devuelve booleano
  { Extrae caracteres de p y c y devuelve verdad si y sólo si
    cada pareja extraída son caracteres iguales. }
  variables sonIguales:booleano
  principio
    sonIguales:=verdad;
    mientrasQue sonIguales and not esPilaVacía(p) hacer
      sonIguales:=cima(p)=primero(c);
      desapilar(p);
      eliminar(c)
    fmq;
    devuelve(sonIguales)
  fin

principio
  cargaFrase(p,c);
  si chequeaPalíndromo(p,c) entonces
    escribir('Es palíndromo')
  sino
    escribir('No es palíndromo')
  fsi
fin
```

Ejercicio: estudiar el coste en tiempo del algoritmo anterior.

Simulación de una cola de espera[1]

Cada día tenemos que **hacer cola** en numerosas ocasiones para obtener un cierto “servicio” por parte de algún agente o “servidor”. No debe sorprendernos, por tanto, que el uso de colas sea importante en muchas aplicaciones informáticas.

Ejemplos de colas se dan en las siguientes situaciones: personas esperando ante ventanillas de bancos o aeropuertos, coches esperando en una calle ante un semáforo en rojo o en una autopista ante un puesto de peaje, procesos generados por los usuario de un sistema informático multiusuario esperando a ser ejecutados por el procesador, llamadas telefónicas recibidas en una centralita esperando hasta obtener línea libre hacia una determinada extensión, piezas de un determinado tipo esperando en un almacén para ser tratadas por una máquina en un sistema de fabricación, etcétera.

Todas las situaciones anteriores pueden ser **simuladas** con un computador utilizando una variable del TAD cola que sirva para **modelar** o imitar la cola que se produce en la realidad y poder responder a preguntas del tipo de: ¿cuánto tiempo tiene que esperar, en media, un cliente para obtener un servicio?, ¿cuál es la longitud media de la cola?, ¿cuál es la varianza de las medidas anteriores?

Vamos a estudiar el caso más sencillo. Hay un solo servidor al que llegan clientes de forma **aleatoria** y el servicio de cada uno de ellos le toma un tiempo fijo al servidor (tiempo de servicio). Los parámetros de entrada para ejecutar una simulación con un computador son: la probabilidad de que durante un intervalo de tiempo de un minuto se produzca la llegada de un cliente, el tiempo (fijo) de servicio a un cliente y la longitud total del intervalo de tiempo que se quiere simular.

Para llevar a cabo la simulación utilizaremos una cola (entendida aquí como una variable del TAD cola) en la que almacenar los clientes esperando para ser servidos. Cada elemento de la cola será, en realidad, el instante (entero no negativo) en el que llegó el cliente.

```
tipo instanteDeLlegada = 0..maxEntero
```

Utilizaremos, además, una variable tiempo (entera no negativa) que modela el **reloj**; su valor inicial será cero y se incrementará de uno en uno, en cada paso de la simulación, contando los minutos transcurridos.

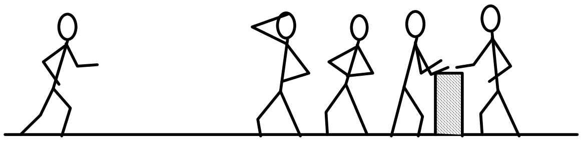
Cuando el cliente llega al principio de la cola, es decir, ante el servidor, la diferencia entre el tiempo en ese momento y el instante de su llegada a la cola es el número de minutos que ese cliente ha esperado en la cola (puede ser cero si cuando llega el cliente la cola está vacía). Podemos sumar todos esos tiempos de espera hasta el final del tiempo total de simulación y dividir la suma por el número de clientes que han llegado, y obtener, así, el tiempo medio de espera de un cliente en la cola.

Para modelar las llegadas aleatorias, utilizaremos una función `random` que nos devuelve un número (pseudo)aleatorio uniformemente distribuido en el intervalo (0,1). Para decidir si un cliente llega o no durante un intervalo de tiempo de un minuto, preguntamos si el valor obtenido con la función random es menor o no que el valor de la probabilidad de llegada de un pasajero en un minuto cualquiera (valor, este último, solicitado como parámetro de entrada antes de comenzar la simulación).

El algoritmo esbozado previamente es como sigue:

```
procedimiento simulador
importa colasDeInstantesDeLlegada
variables cola:colaDeInstantesDeLlegada;
          probabilidadLlegada,esperaMedia:real;
          tiempoServicio,tiempoSimulación,tiempo,
             tiempoQuedaServicio,númeroClientes,
             sumaDeEsperas,instanteLlegada:0..maxEntero;
  función random devuelve real
  { Devuelve un número (pseudo)aleatorio en el intervalo (0,1). }
principio
  escribir('DATOS');
  escribir('Probabilidad de que llegue un cliente durante un minuto:');
  leer(probabilidadLlegada);
  escribir('Tiempo requerido por cada servicio (en minutos):');
  leer(tiempoServicio);
  escribir('Longitud de la simulación (tiempo total, en minutos):');
```

1



```
    leer(tiempoSimulación);

    creaVacía(cola);
    tiempo:=0;
    tiempoQuedaServicio:=0; { el servidor está libre, en principio }
    númeroClientes:=0;
    sumaDeEsperas:=0;

    mientrasQue tiempo ≤ tiempoSimulación hacer
      si random < probabilidadLlegada entonces
        añadir(cola,tiempo)
      fsi;

      si tiempoQuedaServicio = 0 entonces
        si not esVacía(cola) entonces
          instanteLlegada:=primero(cola);
          eliminar(cola);
          sumaDeEsperas:=sumaDeEsperas+(tiempo-instanteLlegada);
          númeroClientes:=númeroClientes+1;
          tiempoQuedaServicio:=tiempoServicio
        fsi
      fsi;

      tiempo:=tiempo+1;

      si tiempoQuedaServicio > 0 entonces
        tiempoQuedaServicio:=tiempoQuedaServicio-1
      fsi
    fmq;

    si númeroClientes = 0 entonces
      esperaMedia:=0.0
    sino
      esperaMedia:=sumaDeEsperas/númeroClientes
    fsi;
    escribir('RESULTADOS');
    escribir('Número de clientes servidos: ',númeroClientes);
    escribir('Tiempo medio de espera (en minutos): ', esperaMedia)
  fin
```

A continuación se muestra el resultado obtenido al ejecutar tres veces el algoritmo anterior para datos idénticos:

```
DATOS
Probabilidad de que llegue un cliente durante un minuto: 0.10
Tiempo requerido por cada servicio (en minutos): 5
Longitud de la simulación (tiempo total, en minutos): 200
RESULTADOS
Número de clientes servidos: 16
Tiempo medio de espera (en minutos): 0.25

DATOS
Probabilidad de que llegue un cliente durante un minuto: 0.10
Tiempo requerido por cada servicio (en minutos): 5
Longitud de la simulación (tiempo total, en minutos): 200
RESULTADOS
Número de clientes servidos: 29
Tiempo medio de espera (en minutos): 3.93

DATOS
Probabilidad de que llegue un cliente durante un minuto: 0.10
Tiempo requerido por cada servicio (en minutos): 5
Longitud de la simulación (tiempo total, en minutos): 200
RESULTADOS
Número de clientes servidos: 19
Tiempo medio de espera (en minutos): 0.68
```

En los tres casos, un cliente llega, en media, cada diez minutos. El tiempo de servicio de cada cliente es de cinco minutos. La observación más interesante sobre los resultados anteriores es la variación de los resultados. Dada la frecuencia de llegada de clientes (uno cada diez minutos) y la longitud de la simulación (200 minutos), el número esperado de clientes servidos debe ser 20, con una cierta varianza alrededor de la media, y en efecto así ocurre. Sin embargo, la gran varianza del tiempo medio de espera en la cola es sorprendente.

El algoritmo anterior puede modificarse de forma que se ejecute la simulación de 200 minutos un total de 200 veces, para el mismo valor de la velocidad de llegadas (1 cliente cada 10 minutos) y para un valor dado del tiempo de servicio. Los resultados que pueden obtenerse de este nuevo algoritmo son, por ejemplo, el máximo tiempo de espera de un cliente (en las 200 simulaciones), la espera media en las 200 simulaciones, la espera media mínima y la espera media máxima de entre las 200 simulaciones.

En la tabla siguiente se muestran los resultados obtenidos ejecutando el algoritmo modificado para distintos valores del tiempo de servicio. Por ejemplo, la primera línea dice que para un tiempo de servicio de 3 minutos, la espera máxima de un cliente en las 200 simulaciones fue de 6 minutos. El tiempo medio de espera fue 0.40 minutos; sin embargo, en (al menos) una simulación de las 200 el tiempo medio de espera fue 0.0 mientras que en otra fue 1.29 minutos.

Tiempo de servicio	**Máxima espera**[1]	**Espera media**	**Mínima espera media**	**Máxima espera media**
3	6	0.40	0.00	1.29
4	11	0.90	0.00	3.35
5	23	1.69	0.00	5.96
6	38	2.81	0.27	16.82
7	48	5.55	0.20	24.00
8	70	7.80	0.28	34.48

Si el tiempo de servicio se aproxima al tiempo medio entre dos llegadas consecutivas de clientes, el sistema se aproxima a un estado de **saturación**. Si la velocidad de llegadas fuese mayor que la de servicio, obviamente el número de clientes en la cola aumentaría *ad infinitum*. Lo más sorprendente vuelve a ser que para un sistema lejos del estado de saturación, la variación de los resultados puede ser grande.

Por ejemplo, para un tiempo de servicio de 5 minutos y los clientes llegando en media cada 10 minutos, al menos un cliente tuvo que esperar en cola durante 23 minutos. En una de las 200 simulaciones, para esos mismo valores de entrada, el tiempo medio de espera en cola fue de 5.96 minutos (superior incluso que el tiempo de servicio), mientras que en otra de las simulaciones el tiempo medio de espera en cola fue de 0.0 minutos (es decir, ninguno de los clientes servidos en los 200 minutos de esa simulación tuvo que hacer cola).

La simulación propuesta en esta lección es la más sencilla que puede plantearse y, de hecho, existen **fórmulas analíticas** que permiten obtener los resultados buscados sin realizar ninguna simulación (por ejemplo, si el tiempo entre dos llegadas consecutivas es una variable aleatoria exponencial de parámetro λ y el tiempo de servicio es otra variable aleatoria exponencial de parámetro μ, entonces el tiempo medio de espera de un cliente en la cola es $W_q=(1/\mu^2)/(1-\lambda/\mu)$. El estudio de modelos de colas como el propuesto aquí y de otros más realistas y complejos, y su análisis, es el objeto de la denominada **Teoría de Colas.**

1