UNIVERSIDAD DE GRANADA

/ UGR / decsai

Departamento de Ciencias de la Computación e Inteligencia Artificial

Estructuras de datos grupo B

Ejercicios sobre pilas, colas y colas con prioridad

`Nota: las cabeceras de las funciones deben ser exactamente tal y como aparecen.`

1. Construye una función a la que se se le pase una pila P de tipo T y dos elementos x,y de tipo T y que modifique la pila de forma que todas las veces que aparezca x se substituya por y (quedando la pila en el mismo estado en el que estaba anteriormente).

   void modificar(stack<T> &P, const T &x, const T &y)

2. Implementa una función para determinar si una expresión contenida en un string tiene una configuración de paréntesis correcta. Debe tener un orden lineal.

   bool parentizada(string expresion)

3. Implementa un TDA cola usando como representación dos pilas.
4. Implementa el TDA pila usando dos colas. ¿Qué orden de eficiencia tienen las operaciones push y pop?
5. Se llama expresión en postfijo a una expresión matemática en la que cada operación aparece con sus dos operandos seguidos por el operador. Por ejemplo: a c + b *

   Escribe un programa que evalúe una expresión en postfijo.

   ab^c*d/e+ donde a = 5, b = 3, c = d = 2, e = 9

   abcde+*+ donde a = 12, b = 4, c = 7, d = 5, e = 2

   ab+cd*+e* donde a = 2, b = 6, c = 3, d = 5, e = 9

   float evalua(string expresion, pair<char, float> *variables, int num_variables)
6. Implementa una función que inserte un elemento en una pila P en la posición

   void insertar(stack<T> &P, int pos, const T &x)

   La pila debe quedar como estaba antes de insertar el elemento (salvo por el nuevo elemento)
7. Implementa una función que inserte un elemento en una cola Q en la posición pos. La cola debe quedar como estaba antes de insertar el elemento (salvo por el nuevo elemento)

   void insertar(queue<T> &P, int pos, const T &x)
8. Usando una pila y una cola, implementa una función que compruebe si un string es un palíndromo.

   bool palindromo(string cadena)
9. Dada una matriz que representa un laberinto, construye una función que determine si se puede llegar desde la entrada hasta la salida. Esta matriz tendrá una ’E’ en la entrada, una ’S’ en la salida, un ’0’ en las casillas por las que se pueda pasar y un ’1’ en las que no. No se puede ir en diagonal.

stack<pair<int,int> > salida_laberinto(char **laberinto, int filas, int columnas)

10. Un tipo ventana es un tipo de dato que permite insertar un elemento, mover derecha, mover izquierda, borrar elemento y devolver elemento. Se implementa usando dos pilas. Implementa ese tipo de dato con las operaciones comentadas.

```
class ventana {
  void insertar(T elemento);
  void mover_derecha();
  void mover_izquierda();
  void borrar_elemento();
  T devolver_elemento() const;
  bool fin() const; // Devuelve true si estamos al final
  bool inicio() const; // Devuelve true si estamos al inicio
}
```

11. Implementa una cola con prioridad de un tipo struct con (apellidos, nombre, prioridad) de forma que los datos salgan de acuerdo a ese tercer campo prioridad.
12. Implementa una cola con prioridad que contenga strings y de la que salgan primero las cadenas de caracteres más largas y que en caso de igualdad salgan por orden alfabético.
13. Implementa una cola con prioridad que contenga strings y de la que salgan primero las cadenas de caracteres que tengan más vocales y que en caso de igualdad salgan por orden alfabético.