

### 1.1. Se solicita

1. Implementar en lenguaje C el algoritmo Búsqueda en profundidad  $DFS(G)$
2. Los experimentos que debe realizar consisten en determinar el costo computacional (tiempo) en recorrer todos los nodos de un grafo.
3. La salida del programa consiste en:
  - Orden de como recorrió todos los nodos del grafo  $G$ .
  - Tiempo en recorrer todos los nodos del grafo  $G$ .
4. Se debe entregar:
  - Código fuente del algoritmo.
  - Informe en  $\text{\LaTeX}$  que contiene:
    - Introducción del problema.
    - Algoritmo implementado.
    - Formulación del experimento.
    - Curvas de desempeño de resultados.
    - Máquina de prueba.
    - Instrucciones de compilación y ejecución.
    - Conclusiones.
5. Normas a cumplir:
  - a) Número de Integrantes: máximo 2.
  - b) Plazo de Entrega:...
  - c) Forma de envío: se envía directorio comprimido tgz, (no se acepta otro formato) a: [rodrigo.abarzua@usach.cl](mailto:rodrigo.abarzua@usach.cl)
6. El nombre del archivo (y directorio) tiene el formato:

$$\sum_{i=1}^2 \text{InicialNombreIntegrante}_i + \text{ApellidoPaterno}_i + \text{InicialApellidoMaterno}_i$$

- Ejemplo: Leoncio Leon Perez y Triston Hiena Perez  $\rightarrow$  LLeonPTHienaP.
- No deber haber ni acentos ni espacios en blanco.



# Matemática Computacional

## Laboratorio 4

Prof: Rodrigo Abarzúa

5 de diciembre de 2013

### 1. Búsqueda Exhaustiva

El término “búsqueda exhaustiva” también se puede aplicar a dos algoritmos muy importantes. Estos recorren todos los vértices ( $V$ ) y aristas ( $E$ ) del grafo  $G = (V, E)$  de manera ordenada. Estos dos algoritmos son los de búsqueda en profundidad ( $DFS(G)$ ) y la búsqueda en anchura ( $BFS$ ). Estos algoritmos han demostrado ser muy útiles para muchas aplicaciones que implican grafos en la inteligencia artificial y la investigación de operaciones. Además, son indispensables para la investigación eficiente de las propiedades fundamentales de los grafos, como la conectividad y la presencia de ciclo.

En este laboratorio estudiaremos Búsqueda en profundidad  $DFS(G)$  que es un algoritmo que permite recorrer todos los nodos de un grafo de manera ordenada. Su funcionamiento consiste en ir expandiendo todos y cada uno de los nodos que va localizando, de forma recurrente, en un camino concreto. Cuando ya no quedan más nodos que visitar en dicho camino, regresa (Backtracking), de modo que repite el mismo proceso con cada uno de los hermanos del nodo ya procesado.

---

#### Algorithm 1 Búsqueda en profundidad $DFS(G)$

---

**Input:** Grafo  $G = (V, E)$

**Output:** Grafo  $G$  con sus vértices marcados con números enteros consecutivos en el orden en que se encontró por primera vez por el recorrido DFS, marcar cada vértice en  $V$  con 0 como marca de ser “no visitado”

```
1:  $count \leftarrow 0$ 
2: for each vertex  $v$  in  $V$  do
3:   if  $v$  is marked with 0 then
4:      $dfs(v)$ 
5:   end if
6: end for
=====FUNCTION  $dfs(v)$ =====
Visita recursivamente todos los vértices no visitados que están conectados con el vértice  $v$  por una arista y los números en el orden en que se encuentran via una variable global  $count \leftarrow count + 1$  marca el vértice  $v$  con  $count$ 
7: for each vertex  $w$  in  $V$  adjacent to  $v$  do
8:   if  $w$  is marked with 0 then
9:      $dfs(w)$ 
10:  end if
11: end for
```

---