

Proyecto 1-Diseño de Algoritmos I

Fabio Castro 10-10132, Leopoldo Pimentel 06-40095

05/02/2015

1 Problema 1

1.1 Análisis de Complejidad

Cuando hablamos del analisis de complejidad de nuestro algoritmo, primero tenemos que ver el analisis de complejidad del brelaz

1.2 Explicación de la solución dada

Nuestra solucion al problema toma en consideracion el paper suministrado sobre el algoritmo brelaz y su modificacion para hacerlo mas optimo. Para implementar dicho algoritmo, usamos el lenguaje C++, con su implementacion de vectors en vez de crear una estructura para almacenar el grafo. De esta forma, tratamos de hacer mas ligera la implementacion en memoria y usamos las herramientas que nos provee el lenguaje.

1.3 PseudoCódigo

Algoritmo 1 Algoritmo de Brelaz para la coloración de grafo modificado

Entrada: G Grafo no dirigido

Salida: *Color* Entero con el número de colores mínimo

```
1: bool back = falso
2: bool block = falso
3:  $k = w + 1$ 
4: marcarClicle( $G$ )
5: mientras cierto hacer
6:   si no back entonces
7:     Determinar  $u_k$  and  $\cup(x_k)$ 
8:     para todo  $c \in \cup(x_k)$  hacer
9:       Determinar el numero de nodos vecinos sin color de  $x_k$  para los que
       no se ha usado el color  $c$ 
10:      Determinar el numero de nodos vecinos sin color de  $x_k$  bloqueados
       en caso de la coloracion de  $x_k$  con  $c$ 
11:      Ordenar los colores  $c \in \cup(x_k)$ . Tomando en cuenta primero el numero
       de bloqueos y luego la cantidad de prevenciones
12:    fin para
13:    si no
14:       $c = \text{color de } x_k$ 
15:       $\cup(x_k) = \cup(x_k) - c$ 
16:      remover label de  $x_k$ 
17:    fin si
18:    si  $\cup(u_k) \neq \emptyset$  entonces
19:       $i = \text{color de orden minimal para } \cup(u_k)$ 
20:      si  $i$  no es un color bloqueado entonces
21:        color del nodo  $x_k = i$ 
22:         $k = k + 1$ 
23:      si  $k > n$  entonces
24:         $q = s$ 
25:        {Se encontro una nueva solucion}
26:      si  $q = w$  entonces
27:        {Salir del ciclo}
28:      fin si
29:       $k = \text{Min entre los } \textit{nodo} \text{ coloreados}$ 
30:      remover todas las labels de los nodos
31:       $\forall x_k, \dots, x_n$  back = cierto
32:    si no
33:      back = falso
34:    fin si
35:  si no
36:    back = cierto
37:    block = cierto
38:  fin si
39: fin si
40: fin mientras
```

Algoritmo 2 Continuacion

```
mientras cierto hacer
  {Continuacion del ciclo pasado}
  si back entonces
    si block entonces
      para todo color bloqueado  $c \in \cup(u_k)$  hacer
        Determinar todos los  $x_c$  nodos bloqueados por  $c$  and aplicar label a
        cada uno de los  $x_c$ 
        block = falso
      fin para
    fin si
    label todos los  $x_k$ 
     $k$  = Maximo de todos los nodos con label
    si  $k \nmid w$  entonces
      {Salir del Ciclo}
    fin si
  fin si
fin mientras
```
