```
#ifndef __GRAPHC_H__
#define __GRAPHC_H__

#define INF    -1
#define NIL    0
#define WHITE  1
#define GREY   2
#define BLACK  3

#include "List.h"

typedef struct Graph{
   ListRef* adj;/*array of lists who's ith element contains
                  neighbors of vertex i */
   int* color; /* color of x = color[x]
                  such that white = 1, grey = 2, black = 3 */
   int* d; /* distance from source to x = d[x] */
   int* P; /* parent of x = P[x] */
   int order; /* # of vertices */
   int size; /* # of edges */
   int source; /* last vertex used by BFS */
}Graph;

typedef struct Graph* GraphRef;

/*** Contructos / Destructors ***/
GraphRef newGraph( int n );
void freeGraph( GraphRef* pG );

/*** Access functions ***/
int getOrder( GraphRef G );
int getSize( GraphRef G );
int getSource( GraphRef G );
int getParent( GraphRef G, int u );
int getDist( GraphRef G, int u );
void getPath( ListRef L, GraphRef G, int u );

/*** Manipulation procedures ***/
void makeNull( GraphRef G );
void addEdge( GraphRef G, int u, int v );
void addArc( GraphRef G, int u, int v );
void BFS( GraphRef G, int s );


/*** Other operations ***/
```



```
void printGraph( FILE* out, GraphRef G );
void printGraphInfo( GraphRef G );

#endif
```