# DANH SACH CAC HAM TRONG Graph:

1- jrb_traverse(node,tree) : Duyet tat ca cac node trong cay tree

2- int getAdjacentVertices (Graph graph, int v, int* output) : Duyet cac dinh ke cua v,luu vao output , return so dinh lien ke

3- int adjacent(Graph graph, int v1, int v2) : Kiem tra xem v1 va v2 co phai le 2 dinh lien ke ko return 0 neu ko,1 neu co

4- void dropGraph(Graph graph) : Xoa do thi

5- void BFS(Graph graph, int start, int stop, void (*func)(int) : Duyet do thi theo chieu rong ,func la ham in dinh ra mac dinh la :void printVertex(int v)

6- void DFS(Graph graph, int start, int stop, void (*func)(int)): Duyet do thi theo chieu sau

7- int indegree (Graph graph, int v, int* output) : Duyet cac dinh u sao cho u->v co lien ket ,luu vao output va tra ve so dinh u thoa man

8- int DAG(Graph graph) : Ham kiem tra xem do thi co vong khong, co thi tra ve 0, khong thi tra ve 1

9- int connected(Graph graph, int start, int stop) : Kiem tra xem 2 dinh start va stop co lien thong voi nhau hay khong, co thi return 1,ko thi 0

## ————————————HAM TRONG JVAL’lib———
## ———————————————

10-Jval new_jval_i(int) : Ham tra ve mot bien moi Jval luu int

11-Jval new_jval_l(long)

12-Jval new_jval_f(float)

13-Jval new_jval_d(double)

14-Jval new_jval_v(/* void */)

15-Jval new_jval_s(char *)

16- int jval_i(Jval); :Cac ham nay tra ve kieu va gia tri tuong ung ma bien Jval luu
long jval_l(Jval);
float jval_f(Jval);
double jval_d(Jval);
void *jval_v(Jval);
char *jval_s(Jval);

## ————————————HAM TRONG JRB’lib———
## ———————————————

17-Cac ham nay dung de insert 1 bien Jval kieu
JRB jrb_insert_str(JRB tree, char *key, Jval val);
JRB jrb_insert_int(JRB tree, int ikey, Jval val);
JRB jrb_insert_dbl(JRB tree, double dkey, Jval val);
JRB jrb_insert_gen(JRB tree, Jval key, Jval val, int

(*func)(Jval,Jval)); /*Ham func la ham compare cua kieu Jval can luu*/

18-Cac ham tim kiem bien Jval trong cay JRB
JRB jrb_find_str(JRB root, char *key);
JRB jrb_find_int(JRB root, int ikey);
JRB jrb_find_dbl(JRB root, double dkey);
JRB jrb_find_gen(JRB root, Jval, int (*func)(Jval, Jval));

19-Cac ham xoa trong JRB
void jrb_delete_node(JRB node);
void jrb_free_tree(JRB root);

**————————————HAM TRONG Dllist'lib————————————————————**

20-
Dllist new_dllist(); Tao mot dllist
free_dllist(Dllist); Xoa Dllist
dll_append(Dllist, Jval); Insert vao cuoi
dll_prepend(Dllist, Jval); Insert vao dau
dll_insert_b(Dllist, Jval); Insert truoc node
dll_insert_a(Dllist, Jval); Insert sau node
dll_delete_node(Dllist); Xoa node
int dll_empty(Dllist); Del Dllist
Dllist dll_first(Dllist); Lay node dau
Dllist dll_last(Dllist); Lay node cuoi

A