

CSE 101

期中考试1复习问题解答

1. (20分)仅使用pa1的项目描述中定义的List ADT操作，编写一个客户函数，标题为

```
bool isPalindrome(List L)
```

如果 L 所代表的整数序列是一个调色板（即与它自己的反面相同），你的函数将返回真，如果 L 不是一个调色板，将返回假。

几个可能的解决方案之一。

```
bool isPalindrome(List L) {  
  
    bool eq;  
    列表R = newList();  
    for (moveFront(L); index(L) >= 0; moveNext(L)) {  
        prepend(R, get(L));  
    }  
    eq = equals(L, R);  
    freeList(&R);  
  
    返回eq。  
}
```

另一个解决方案。

```
bool isPalindrome(List L) {  
  
    bool eq = true;  
    List C = copyList(L);  
    while (eq && length(C) > 1) {  
        eq = (front(C) == back(C));  
        deleteFront(C);  
        deleteBack(C);  
    }  
    freeList(&C);  
  
    返回eq。  
}
```

2. (20分)仅使用pal的项目描述中定义的List ADT操作，编写一个客户函数，标题为

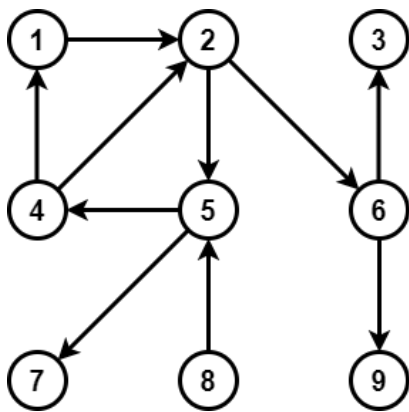
空白 替换(List L, int x, int y)

你的函数将用y替换L中所有出现的x。如果x不在L中，你的函数将不会对L中的整数序列做任何改变。

几个可能的解决方案之一。

```
void Replace(List L, int x, int y){  
    for(moveFront(L); index(L)>=0; moveNext(L)){  
        如果( get(L)==x  
            ){  
            set(L, y).  
        }  
    }  
}
```

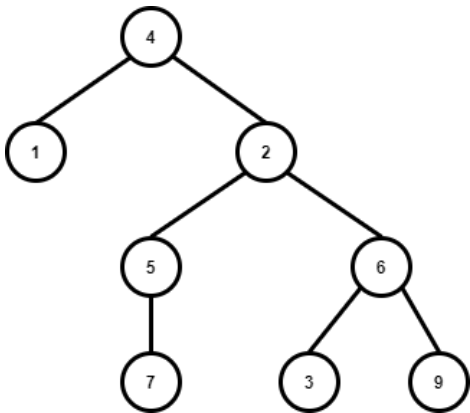
3. (20分)在下图的数字图上运行BFS算法，以顶点4为源。在表格中填入邻接列表、颜色、与源的距离以及BFS树中的父母。将发现的顶点按其进入队列的顺序列出。画出结果的BFS树。



顶点	暂停	颜色	距离	父母
1	2	黑	1	4
2	5 6	黑	1	4
3		黑	3	6
4	1 2	黑	0	无
5	4 7	黑	2	2
6	3 9	黑	2	2
7		黑	3	5
8	5	白色	宇宙空	无

排队。 4 1 2 5 6 7 3 9

BFS树。



4. (20分)给定一个连通(无定向)图 G , G 的直径是 G 中任何两个顶点 x 和 y 之间可能的最大距离, 即。

$$\text{diameter}(G) = \max\{ \delta(x, y) \mid x, y \in V(G) \}$$

只使用项目描述中为pa2定义的Graph ADT函数, 编写一个客户端函数, 标题为

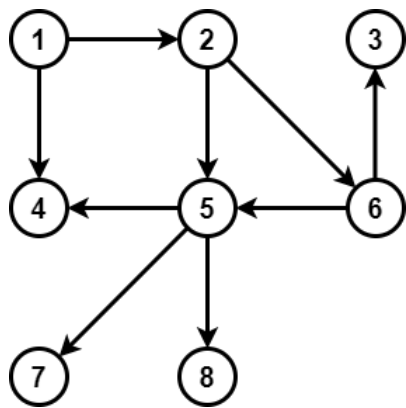
```
int diameter(Graph G)
```

你的函数将计算并返回其输入图 G 的直径。

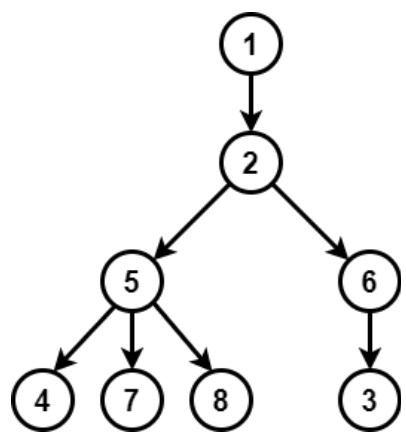
几个可能的解决方案之一。

```
int diameter(Graph G){  
  
    int max, d, x, y;  
  
    max = 0;  
    for (x=1; x<=getOrder(G); x++){  
        BFS(G, x);  
        for(y=1; y<=getOrder(G); y++){  
            d = getDist(G, y);  
            如果( d>max ){  
                max = d。  
            }  
        }  
    }  
    返回最大。  
}
```

5. (20分) 在下图的数字图上运行DFS算法。在DFS()的主循环中通过增加顶点标签来处理顶点。在Visit()的for循环中通过增加顶点标签来处理顶点。当顶点处理完后，把它们推到一个堆栈中。在下面的表格中填写邻接列表的表示方法、发现时间、完成时间和DFS森林中的父母。画出所产生的DFS森林，并显示DFS完成时堆栈的状态。将所有的边分类为树型、后退型或交叉型。



顶点	暂停	发现	结束	父母
1	2 4	1	16	无
2	5 6	2	15	1
3		12	13	6
4		4	5	5
5	4 7 8	3	10	2
6	3 5	11	14	2
7		6	7	5
8				



DFS森林。

堆栈。 1 2 6 3 5 8 7 4

边缘分类。

树： (1, 2) (2, 5) (2, 6) (5, 4) (5, 7) (5, 8) (6, 3)

后面。前进：

(1, 4) 。

十字架: (6, 5)