**BLACK BOX TESTING**

**Equivalence Class Testing and Boundary Value Testing**

User must be able to search and read a book

|  |  |
| --- | --- |
| Parameter | Equivalence Class (Valid and Invalid) |
| Keyword Length | One valid EC: {1 <= x <= 99}  Two invalid ECs: {x <= 0}, {x >= 100} |
| Book Format | One valid EC: {epub}  No invalid EC. |

Boundary Value Testing

|  |  |
| --- | --- |
| Parameter | Equivalence Class (Valid and Invalid) |
| Keyword\_Length | Lower Boundary 1: 0, 1, ~~2~~  Upper Boundary 99: ~~98~~, 99, 100 |
| Book\_Format | N/A |

VALID Keyword\_Length(1, 99)

|  |  |  |  |
| --- | --- | --- | --- |
| Keyword\_Length | Book\_Format | Expected Result | Tested Result |
| 1 | epub | The app loads the book to epub viewer. | The app loads the book to epub viewer. |
| 99 | epub | The app loads the book to epub viewer. | The app loads the book to epub viewer. |

INVALID Keyword\_Length(0, 100)

|  |  |  |  |
| --- | --- | --- | --- |
| Keyword\_Length | Book\_Format | Expected Result | Tested Result |
| 0 | epub | The app waits for input | The app waits for input |
| 100 | epub | The app shows the no result page. | The app shows the no result page. |

**WHITE BOX TESTING**

**Basis Path Testing**

**Show all thoughts**

protected void onCreate(Bundle savedInstanceState){

1 super.onCreate(savedInstanceState);

2 setContentView(R.layout.activity\_thought);

3 SQLiteDatabase db = dbHelper.getWritableDatabase();

4 Cursor cursor = db.query("Post", null, null, null, null, null, null);

5 if(cursor.moveToFirst()){

6 do{

7 int userID = cursor.getInt(cursor.getColumnIndex("userID"));

8 String userName = cursor.getString(cursor.getColumnIndex("userName"));

9 String postTitle = cursor.getString(cursor.getColumnIndex("postTitle"));

10 String postDesc = cursor.getString(cursor.getColumnIndex("postDesc"));

11 newPost = new Post(userID, userName, postTitle, postDesc);

12 newPost.setPostID(cursor.getInt(cursor.getColumnIndex("id")));

13 postList.add(newPost);

14 Log.d("PostActivity", "userName is " + userName);

15 Log.d("PostActivity", "postTitle is " + postTitle);

16 Log.d("PostActivity", "postDesc is " + postDesc);

17 }while(cursor.moveToNext());

18 cursor.close();

19 }

20 RecyclerView recyclerView = (RecyclerView) findViewById(R.id.recycler\_view);

21 LinearLayoutManager layoutManager = new LinearLayoutManager(this);

22 recyclerView.setLayoutManager(layoutManager);

23 recyclerView.setHasFixedSize(true);

24 adapter = new PostAdapter(postList);

25 recyclerView.setAdapter(adapter);

26 swipeRefresh = (SwipeRefreshLayout) findViewById(R.id.swipe\_refresh);

27 swipeRefresh.setColorSchemeColors(Color.GREEN);

28 swipeRefresh.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {

29 @Override

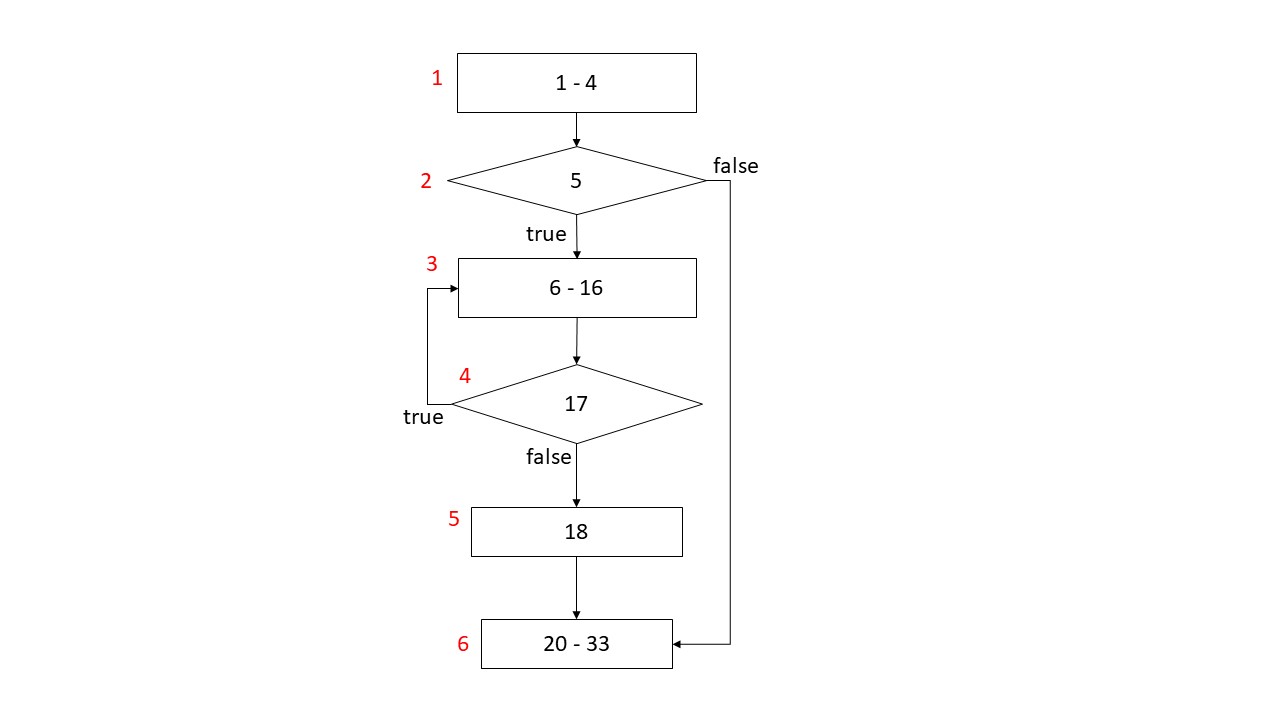
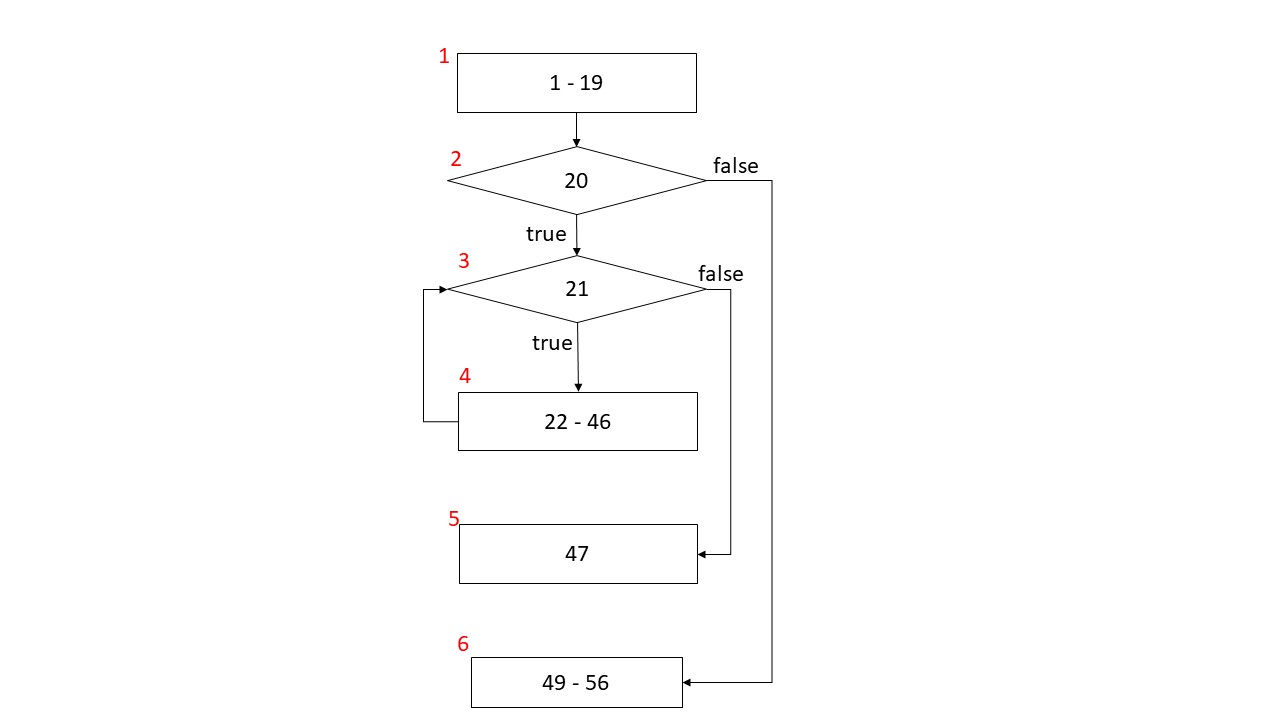
30 public void onRefresh() {

31 refreshPosts();

32 }

33 });

}



CC = |decisionpoint| + 1 = 3

|  |
| --- |
| * Three basis paths  1. 1, 2, 3, 5 2. 1, 2, 3, 4, 5 3. 1, 2, 6 |
| * Two test cases  1. Infeasible basis path 2. Two thoughts are posted 3. No thoughts are posted |
| * Real execution paths  1. – 2. 1,2,3,4,3,4,3,5 3. 1,2,6 |

**Search book**

protected void onCreate(Bundle savedInstanceState) {

1 super.onCreate(savedInstanceState);

2 setContentView(R.layout.activity\_search\_by\_bookname);

3 String keyword= getIntent().getStringExtra("searchedWord");

4 Search search = new Search();

5 String content = "";

6 try{

7 content = search.getContent("{\"book\_title\":\""+keyword+"\"}");

8 } catch(IOException e){

9 e.printStackTrace();

10 }

11 List<BookInfo> resultBookList=new ArrayList<>();

12 try{

13 resultBookList = search.parseJson(content);

14 } catch (Exception e) {

15 e.printStackTrace();

16 }

17 int size = resultBookList.size();

18 final TableLayout layout = (TableLayout) findViewById(R.id.seachResultTable);

19 TableRow[] tableRows = new TableRow[size];

20 if(size!=0){

21 for(int i = 0; i < size; i++){

22 final BookInfo book = resultBookList.get(i);

23 tableRows[i] = new TableRow(this);

24 ImageView cover = new ImageView(this);

25 cover.setImageResource(R.drawable.book\_icon);

26 new ImageLoadTask(book.getCoverPage(), cover).execute();

27 TableRow.LayoutParams cover\_param = new TableRow.LayoutParams(250,250);

28 cover\_param.setMargins(60,90,0,0);

29 cover.setLayoutParams(cover\_param);

30 TextView title = new TextView(this);

31 title.setText(book.getBookTitle());

32 TableRow.LayoutParams title\_param = new TableRow.LayoutParams(500,250);

33 title\_param.setMargins(60,90,0,0);

34 title.setLayoutParams(title\_param);

35 tableRows[i].addView(cover);

36 tableRows[i].addView(title);

37 layout.addView(tableRows[i]);

38 tableRows[i].setOnClickListener(new View.OnClickListener() {

39 @Override

40 public void onClick(View v) {

41 Intent toBookDetail = new Intent(SearchByBookName.this, BookDetailActivity.class);

42 toBookDetail.putExtra("BookInfo", Parcels.wrap(book));

43 startActivity(toBookDetail);

44 finish();

45 }

46 });

47 }

48 } else{

49 TableRow tableRow = new TableRow(this);

50 ImageView no\_result = new ImageView(this);

51 TableRow.LayoutParams layoutParams = new TableRow.LayoutParams(1200,1400);

52 no\_result.setLayoutParams(layoutParams);

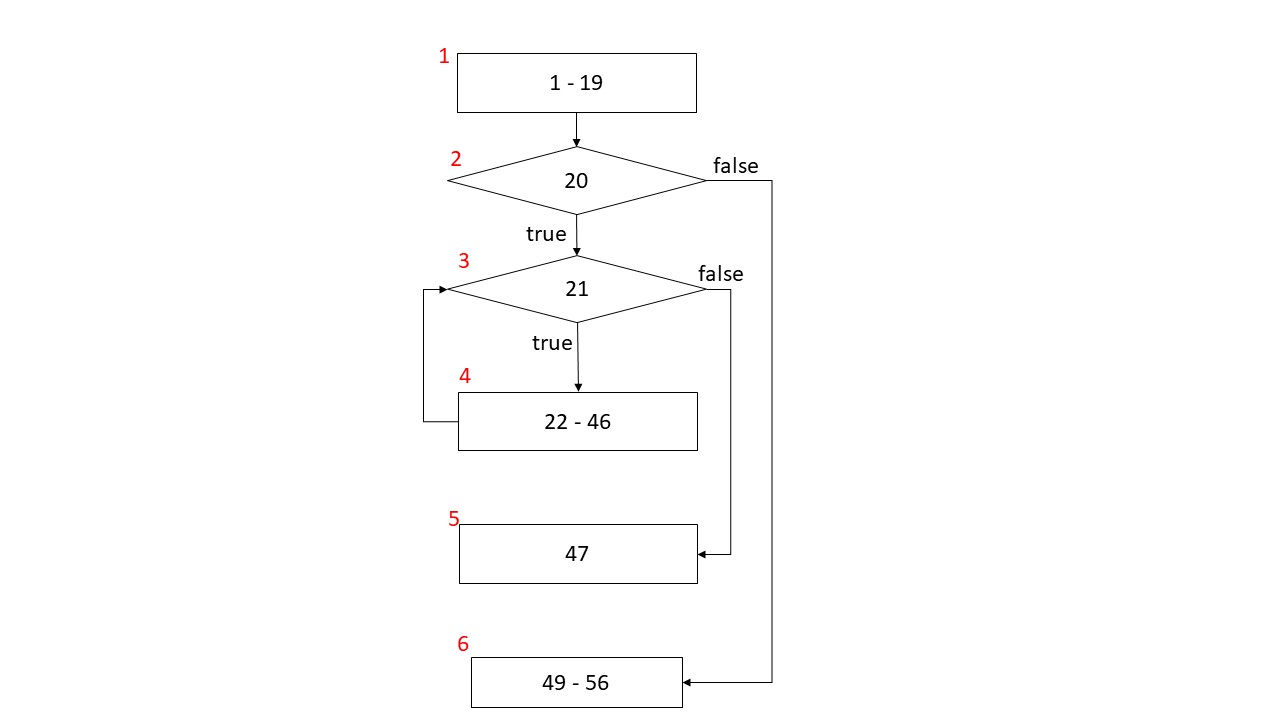
53 tableRow.addView(no\_result);

54 layout.addView(tableRow);

55 no\_result.setImageResource(R.drawable.no\_results\_found);

56 }

}



CC = |decisionpoint| + 1 = 3

|  |
| --- |
| * Three basis paths  1. 1, 2, 3, 5 2. 1, 2, 3, 4, 5 3. 1, 2, 6 |
| * Three test cases  1. Infeasible basis path 2. “Tanjong Rhu” 3. “Boy Town” |
| * Real execution paths  1. - 2. 1,2,3,4,5 3. 1,2,6 |

Thought fragment

public void onViewCreated(View view, Bundle savedInstanceState){

1 super.onViewCreated(view, savedInstanceState);

2 SQLiteDatabase db = dbHelper.getWritableDatabase();

3 Cursor cursor = db.rawQuery("SELECT \* FROM POST order by id desc", null);

4 if(cursor!=null){

5 if(cursor.moveToFirst()){

6 do{

7 int userID = cursor.getInt(cursor.getColumnIndex("userID"));

8 String userName = cursor.getString(cursor.getColumnIndex("userName"));

9 String postTitle = cursor.getString(cursor.getColumnIndex("postTitle"));

10 String postDesc = cursor.getString(cursor.getColumnIndex("postDesc"));

11 int postID = cursor.getInt(cursor.getColumnIndex("id"));

12 newPost = new Post(userID, userName, postTitle, postDesc);

13 newPost.setPostID(postID);

14 postList.add(newPost);

15 Log.d("MainActivity", "userName is " + userName);

16 Log.d("MainActivity", "postTitle is " + postTitle);

17 Log.d("MainActivity", "postDesc is " + postDesc);

18 }while(cursor.moveToNext());

19 cursor.close();

20 }}

21 adapter = new PostAdapter(postList);

22 recyclerView.setAdapter(adapter);

23 swipeRefresh = (SwipeRefreshLayout) view.findViewById(R.id.swipe\_refresh);

24 swipeRefresh.setColorSchemeColors(Color.GREEN);

25 swipeRefresh.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {

26 @Override

27 public void onRefresh() {

28 refreshPosts();

29 }

30 });

31 dbHelper = new MyDbHelper(getActivity(), "Post.db", null, 1);

32 ImageView plussign = (ImageView)view.findViewById(R.id.plussign);

33 plussign.setOnClickListener(new OnClickListener() {

34 @Override

35 public void onClick(View v) {

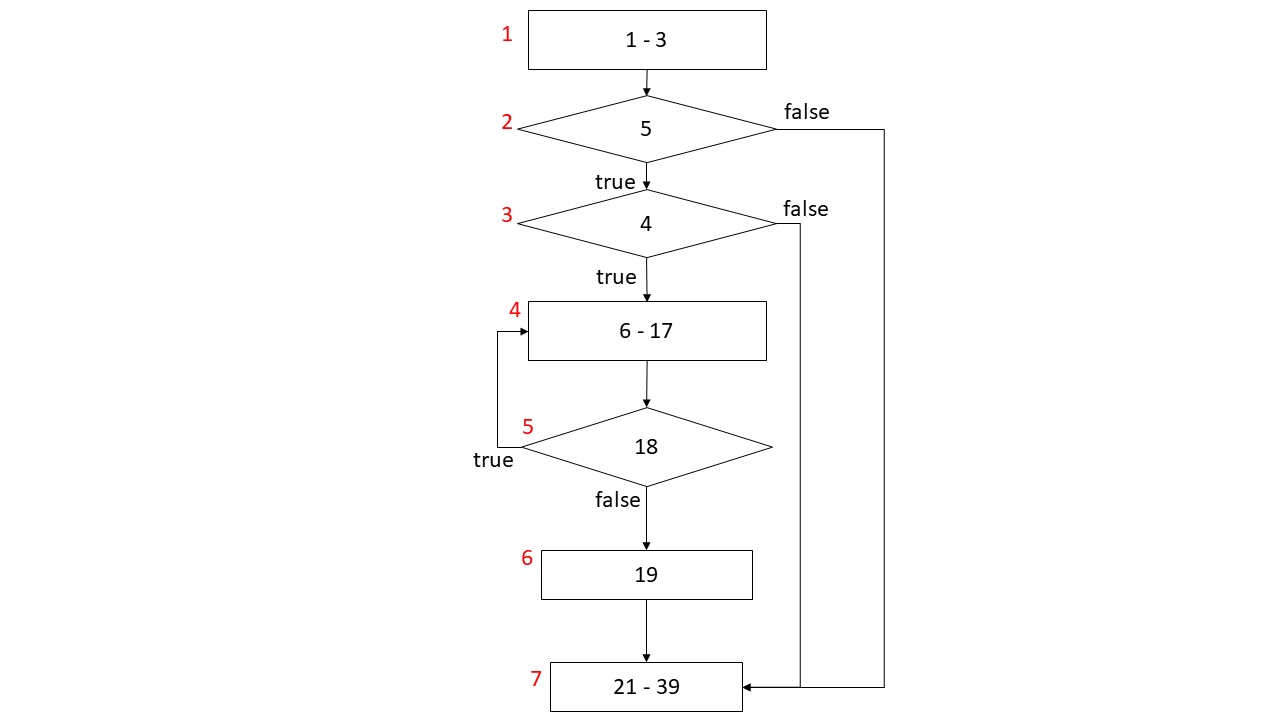
36 Intent toPost = new Intent(getActivity().getApplication(), PostActivity.class);

37 startActivity(toPost);

38 }

39 });

}



CC = |decisionpoint| + 1 = 4

|  |
| --- |
| * Four basis paths  1. 1, 2, 3, 4, 5, 6 2. 1, 2, 3, 4, 5, 6 3. 1, 2, 7 4. 1, 2, 3, 7 |
| * Four test cases  1. Only one thought in database 2. Three thoughts in database 3. No thoughts in database 4. Infeasible basis path |
| * Real execution paths  1. 1,2,3,4,5,6,7 2. 1,2,3,4,5,4,5,4,5,6,7 3. 1,2,7 4. - |