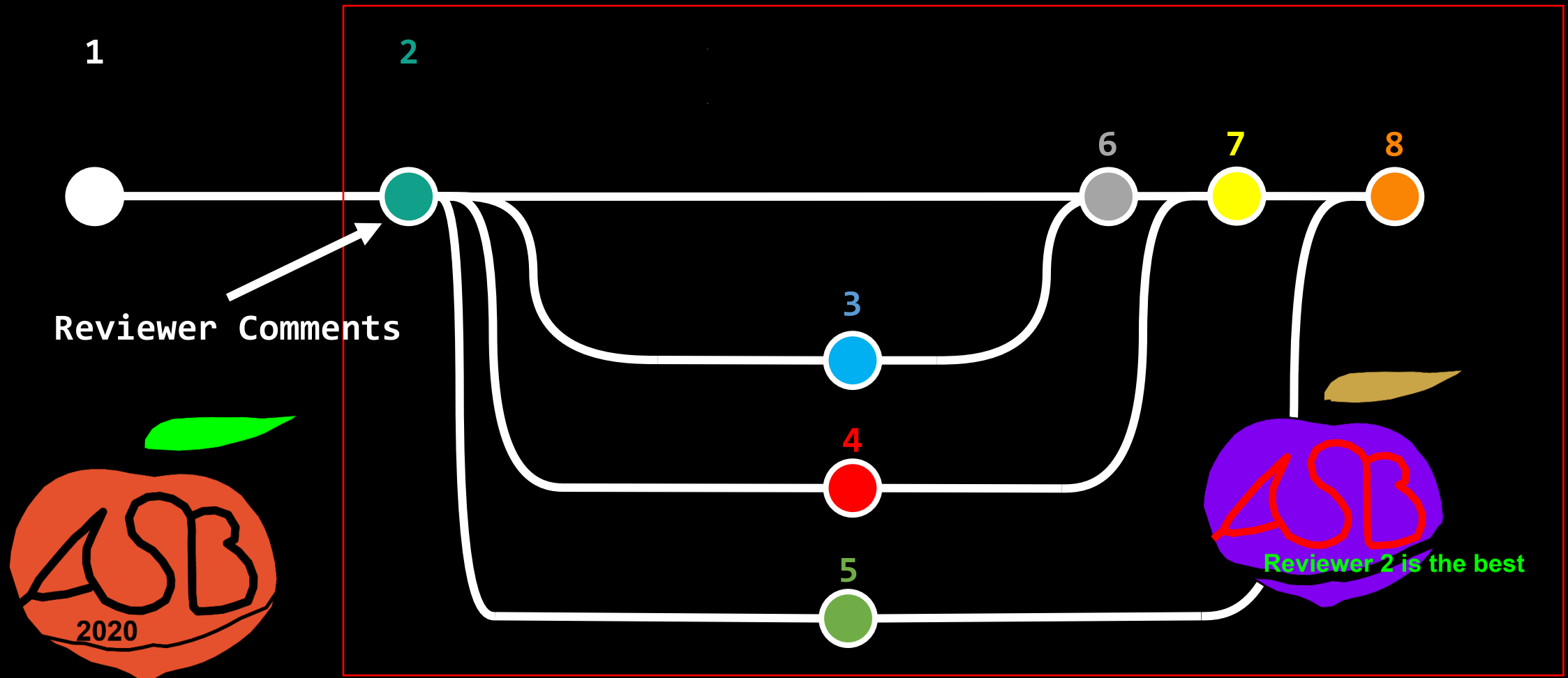
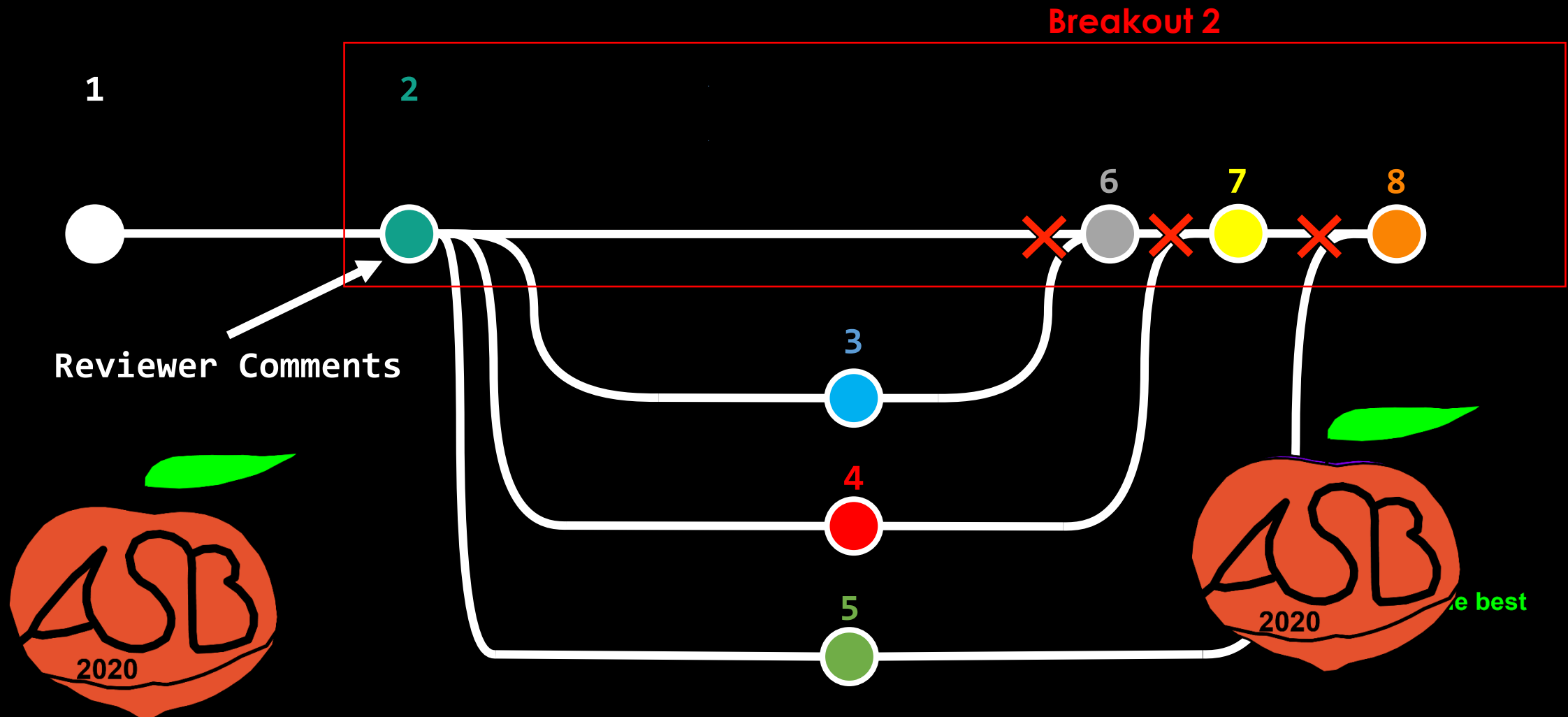


Recap

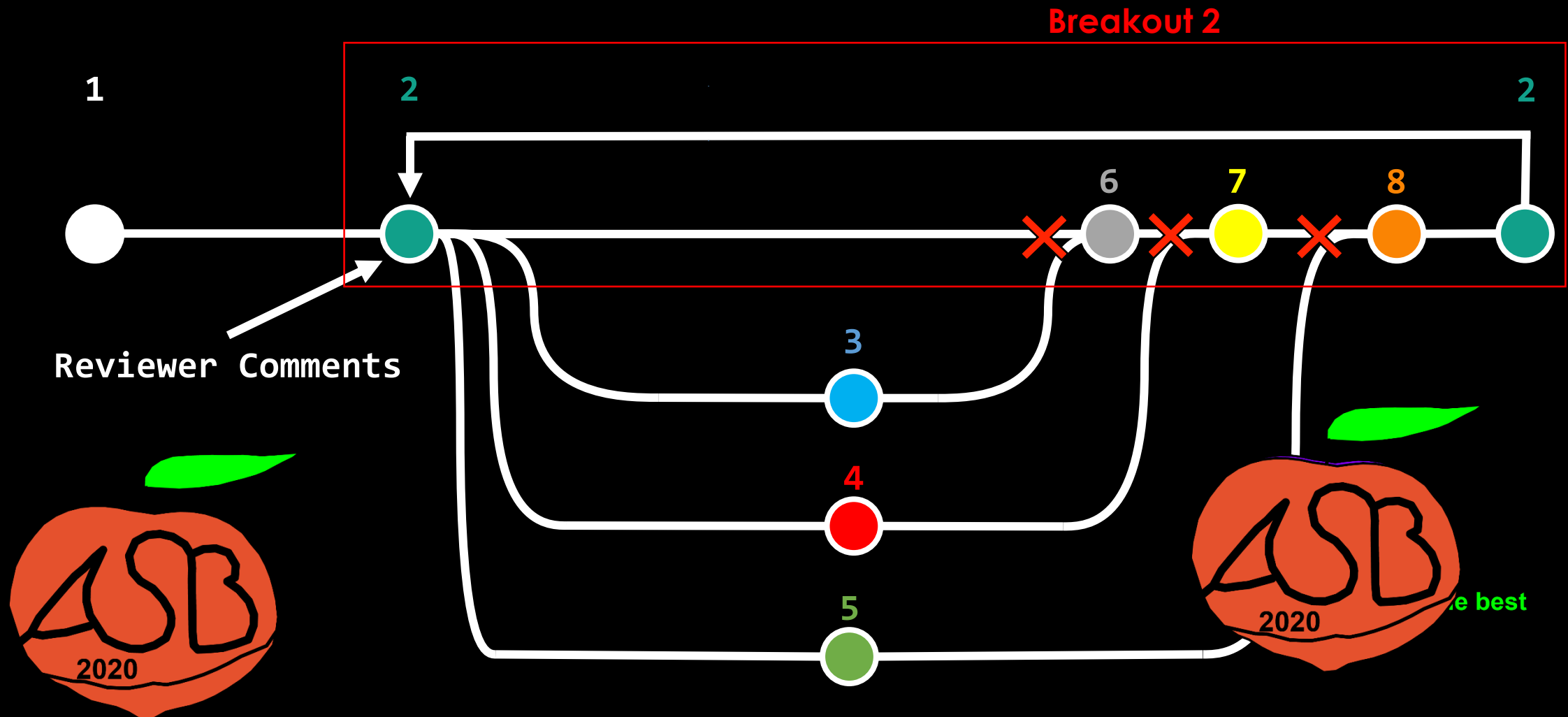
Breakout 1



Recap

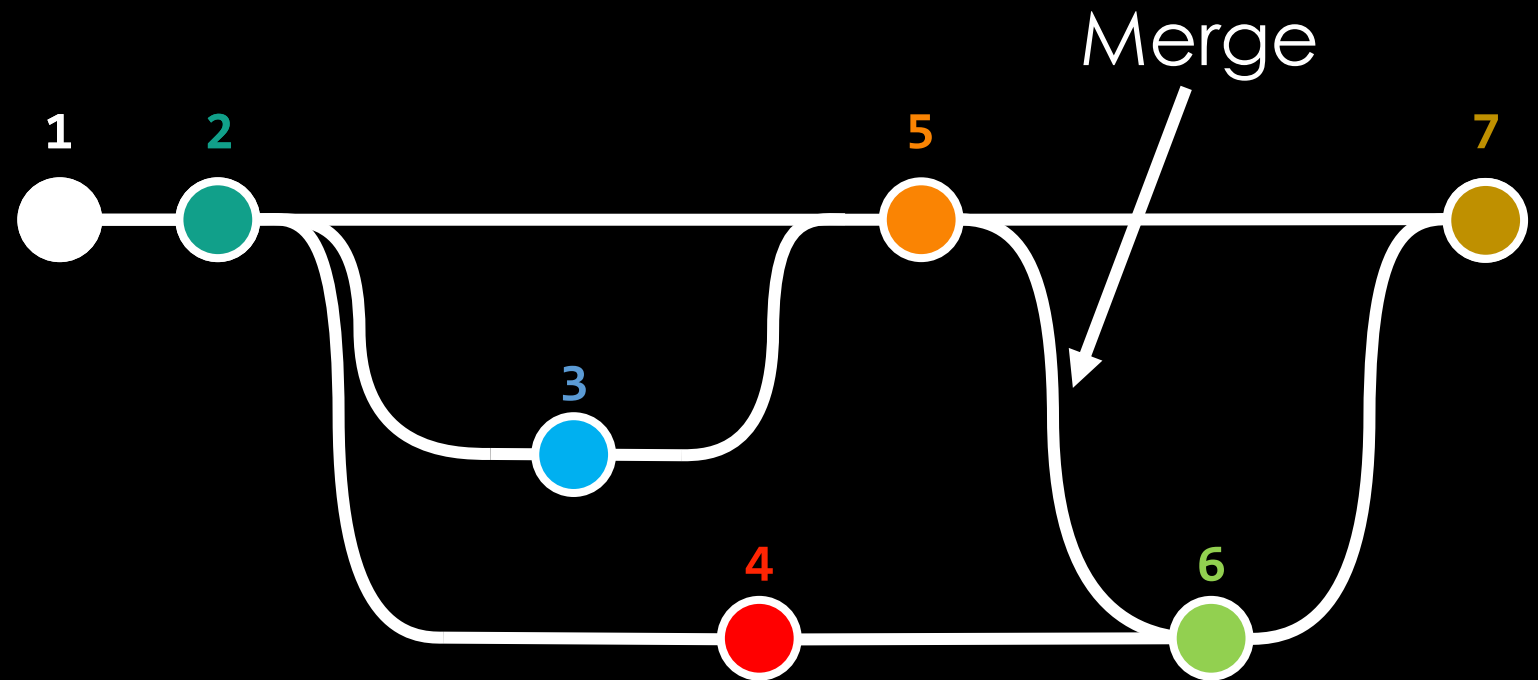


Recap



Extra Git Functions

git merge branch



*git rebase is another way to incorporate changes in main branch to feature branch

Extra Git Functions

git merge branch

git revert commit

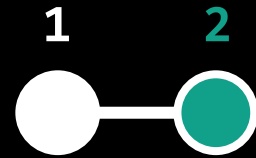


Extra Git Functions

git merge branch

git revert commit

git checkout commit



Checkout

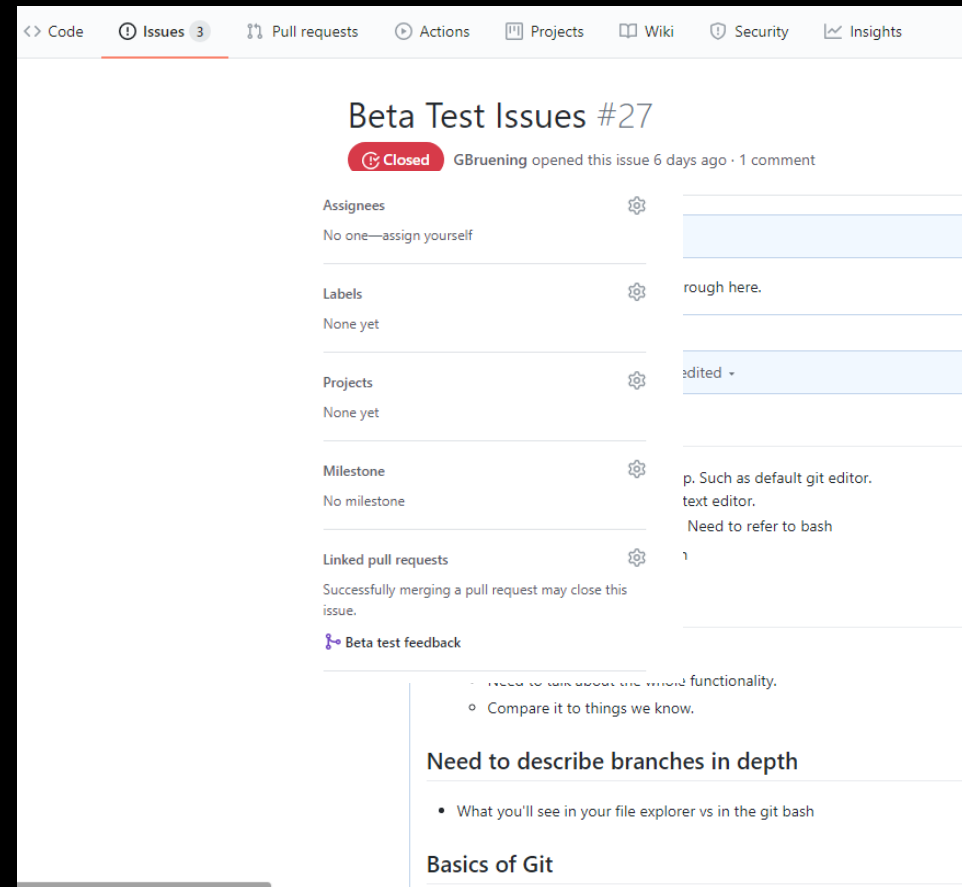


Additional Considerations

Issue Tracking

- Assigning Tasks

Commit History



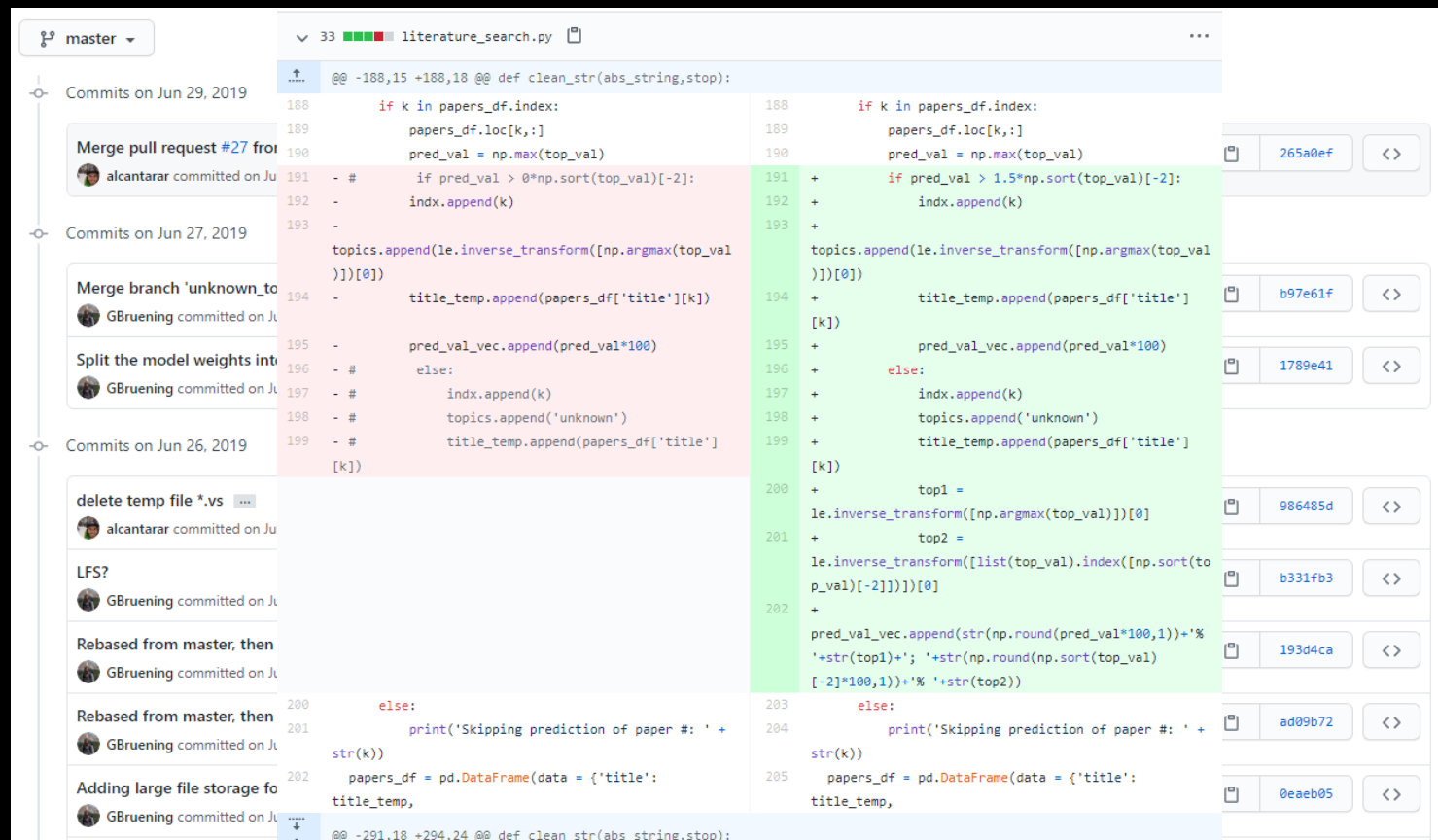
Additional Considerations

Issue Tracking

- Assigning Tasks

Commit History

- Viewing changes



The screenshot displays a GitHub pull request interface. On the left, the 'Commits' tab is active, showing a list of commits from June 26, 2019, to June 29, 2019. The commits include 'Merge pull request #27 from...', 'Merge branch 'unknown' to...', 'Split the model weights into...', 'delete temp file *.vs', 'LFS?', 'Rebased from master, then', and 'Adding large file storage for...'. The main area shows a diff for the file 'literature_search.py'. The diff highlights changes in the 'clean_str' function, specifically in the 'if k in papers_df.index:' block. The changes include adding a condition 'if pred_val > 1.5*np.sort(top_val)[-2]:' and modifying the 'pred_val' calculation. The right side of the diff shows the original code, and the left side shows the changes. The diff is color-coded: green for additions and red for deletions. The commit hash '265a0ef' is visible on the right.

```
@@ -188,15 +188,18 @@ def clean_str(abs_string, stop):
188     if k in papers_df.index:
189         papers_df.loc[k, :]
190         pred_val = np.max(top_val)
191     - # if pred_val > 0*np.sort(top_val)[-2]:
192     -     indx.append(k)
193     -
194     topics.append(1e.inverse_transform([np.argmax(top_val)])[0])
195     - title_temp.append(papers_df['title'][k])
196     -
197     pred_val_vec.append(pred_val*100)
198     - # else:
199     -     indx.append(k)
200     - # topics.append('unknown')
201     - # title_temp.append(papers_df['title']
202     [k])
203
204     + if pred_val > 1.5*np.sort(top_val)[-2]:
205     +     indx.append(k)
206     +
207     + topics.append(1e.inverse_transform([np.argmax(top_val)])[0])
208     + title_temp.append(papers_df['title']
209     [k])
210     +
211     + pred_val_vec.append(pred_val*100)
212     +
213     + else:
214     +     indx.append(k)
215     +     topics.append('unknown')
216     +     title_temp.append(papers_df['title']
217     [k])
218     +
219     + top1 =
220     + 1e.inverse_transform([np.argmax(top_val)])[0]
221     + top2 =
222     + 1e.inverse_transform([list(top_val).index([np.sort(to
223     p_val)[-2])])[0])
224     +
225     + pred_val_vec.append(str(np.round(pred_val*100,1))+ '%
226     '+str(top1)+'; '+str(np.round(np.sort(top_val)
227     [-2]*100,1))+ '% '+str(top2))
228     +
229     + else:
230     +     print('Skipping prediction of paper #: ' +
231     str(k))
232     +     papers_df = pd.DataFrame(data = {'title':
233     title_temp,
234     @@ -291,18 +294,24 @@ def clean_str(abs_string, stop):
```