CS304 MP1 Part 1 Tool-supported fixing *(Total: 10 points)*

Deadline: March 24 **(Tuesday), 11.59pm**

The goal of this MP is for you to try using automated program repair to help you in fixing bugs for your project. In this part, you will run one semi-automated tool (Crossfix) for fixing your open GitHub issue. Read the tutorial on lab-5 about how to run. Accept the invitation link for part 1: https://classroom.github.com/a/XBHhklHo

What to submit?

- **sampled.csv** with the labeled table. You should include the label for each recommended closed issue link. *(-1 points for each missing label, Total points: 10 points)*
- **README.md** with **your name and id**

Select one GitHub issue $i_{selected}$ that you will implement. Use Crossfix to find the relevant closed issue that you can learn from. In Crossfix, the issue $i_{selected}$ is called the *query issue* because it is used to find other relevant issues. For each query issue link, the tool extracts multiple keywords and add filter words to search for closed issue on GitHub. **Note that Crossfix will give you a list of relevant issues and you need to make each of the relevant issue (every row in the csv).** From you experiences, mark these keywords by “useful” or “useless”, and give the suggested query keywords combination if “useless”.

For each closed issue,
1) please give the short label of usefulness: “Y” (is useful) or “N”,
2) If there are some items labeled with “Y”, give the top 10 ranks from the most useful (1st) to the less useful,
3) corresponding useful (for query issue) level: “fix”, “debug”, “not

relevant”, “no result”,

Sample of labeled CSV

query_url	keywords	close_url	usefulness	rank_usefulness	label_usefulness
https://github.com/ankidroid/Anki-Android/issues/6432	crash card browser multi-select mode edit change type in:title	https://github.com/ankidroid/Anki-Android/issues/6432	N		fix
https://github.com/ankidroid/Anki-Android/issues/6432	crash card browser multi-select mode edit change type	https://github.com/ankidroid/Anki-Android/issues/6432	Y	2	debug
https://github.com/ankidroid/Anki-Android/issues/6432	crash card browser	https://github.com/ankidroid/Anki-Android/issues/2746	N		not relevant

	multi-select mode edit change type				
https://github.com/ankidroid/Anki-Android/issues/6432	crash card browser multi-select mode edit change type	https://github.com/termux/termux-app/issues/1072	Y	1	debug
https://github.com/ankidroid/Anki-Android/issues/6432	crash card browser		N		no result

CS304 MP1 Part 2 Tool-supported fixing *(Total: 25 points)*

Deadline: April 14 **(Tuesday), 11.59pm**

The goal of this MP is for you to try using automated program repair to help you in fixing bugs for your project. In this part, you will run two automated program repair tools (Astor and Prapr). **Note that you may need to spend more than 24 hours in installing and running all these tools.** Read the tutorial on lab-5 about how to run. Accept the invitation link for part 2 at:

https://classroom.github.com/a/RY-J75Dp

What to submit?

- **README.md** with **your name and id.** For each open issue, answer the question for APR below. **Put all answers in README.md.**
- For each open issue,
  Astor: you should upload exception stack trace or success fixed patch (in git diff mode), check terminal output for this information
  Prapr: you should upload exception stack trace or success fixed patch (in html mode), check target/prapr-reports/2021XXXXXXX folder within the buggy project for this information
  If astor or prapr successfully fix a buggy program, you should also submit the time it takes from the output information or fix report
- Question 3 (5 points): Answer the questions in the "Tools supported fixing" survey in Sakai under the "Tests and Quiz". **Answer the questions only after running all the three tools (Crossfix, Astor, Prapr).**

1. Select one GitHub issue that you will implement. **Write at least one failing test for the selected issue.** Answer the following question after running Astor (set the timeout to be 12 hours).

   a) What is the link for the selected issue? *(1 point for adding link)*

b) How long did it took for you to run it before the patch generation completes (patch succesfully generated or timeout within 12 hours) ? *(1 point for adding time)*

c) From you experiences of running the tool, mark the status by “Can run” or “Can't run”. *(1 point for adding status)*

d) If the status is "Can run", please then specify whether Astor or Prapr can repair it. **Note that Astor can support Maven and Gradle projects (If you select a gradle project, as Astor doesn't support Gradle well and if you cannot run Astor with gradle, you can choose to select issue from the second project in your group). If your answer is "you didn't run because of you have selected gradle project", you will not get any score. You need to show screenshot of you running Astor and its output.** If yes and a patch is generated, please provide the generated patch, and answer the question "Do you think that the generated patch is correct?". If yes and a patch is not generated, write down the reason that the bug cannot be fixed. The possible reasons can be:

i. The tool cannot find a plausible patch in the search space

ii. The tool cannot explore the whole search space in the given timebound

If the status is "Can't run", please provide your reasons and attach with a screenshot., we have prepared several optional reasons for you, and you can write the special problem you encountered. *(2 points for adding the question on correctness/reason for failure, 5 points for adding the successful patch/failure screenshot, -7 points if didn't run the tool.)*

i. Astor cannot parse the program correctly, example exception:

(1) at spoon.support.compiler.jdt.*

ii. Fault localization cannot work properly, example exception:

(2) "main" java.lang.NullPointerException at com.gzoltar.core.GZoltar.run

iii. Astor doesn't support the java version

(3) UnsupportedClassVersionError

2. Select the same GitHub issue as Question 1 GitHub issue. You also need **at least one failing test for Prapr. Make sure that your new test can be run with mvn test (This is how Prapr find all tests in the test suite).** Answer the following question after running Prapr (set the timeout to be 12 hours).

a) What is the link for the selected issue? *(1 point for adding link)*

b) How long did it took for you to run it before the patch generation completes (patch succesfully generated or timeout within 12 hours) ? *(1 point for adding time)*

c) From you experiences of running the tool, mark the status by “Can run” or “Can't run”. ***(1 point for adding status)***

d) If the status is "Can run", please then specify whether Prapr can repair it. **Note that Prapr has good support maven project but not so much for Gradle project (If you select a gradle project, you can choose to select issue from the second project in your group). If your answer is "you didn't run because of you have selected gradle project", you will not get any score. You need to show screenshot of you running Prapr and its output.** If yes and a patch is generated, please provide the generated patch, and answer the question "Do you think that the generated patch is correct?". If yes and a patch is not generated, write down the reason that the bug cannot be fixed. The possible reasons can be:

i. The tool cannot find a plausible patch in the search space

ii. The tool cannot explore the whole search space in the given timebound

If the status is "Can't run", please provide your reasons and attach a screenshot showing the failure to run. If you have encountered an exception/error, please write the exception/error message. ***(2 points for adding the question on correctness/reason for failure, 2 points for adding the successful patch/failure screenshot)***

3. Survey (5 points):
Answer the question about the usefulness of different tools (Tool-supported fixing) in the survey in Sakai.