Algorithm initial ideas

- The focus for the algorithm will be to capture and implement spaced repetition.
- Space repetition is a form of studying proving to significantly help transferring information into long term memory (longer retention of information)
- "level" is how well the user knows with a given flashcard. The higher the level, the longer it will take for the flashcard to appear for revision.
- A user needs to practice any flashcard

$$\sum 0n2n \sum 0n2n$$

. Where "n" is "

$$0<n<60<n<6$$

" and "level - 2". **"0"** is the first time a user sees a card **for revision**. E.g.: the 4th time a user will be seeing a flashcard will be on day:

$$1+2+4 = 71+2+4 = 7$$

, (day 0 = first time, day 1= revision 1, day 3= revision 2, day 7 = revision 3, total: 4). Here, level will be 4, and the flashcard is revised 3 times.

- Users can rate questions and the algorithm will adapt accordingly.
- Default: user gives a rating "easy" to a flashcard each revision time.
- The maximum revision, for each flashcard, will be 5 times. Meaning a flashcard can have max level "7" (first time is not revision and considered level "1"). for this to happen, a flashcard must be given an easy rating each revision time. Then the given flashcard will be assigned as "mastered" and only visited once every 31 days for revision. (31 because the next repetition would be 63 which is too long)
- The ratings system is not finalized but for now, assume only 3 ratings: Easy, Challenging, Hard
- In an event where "challenging" or "hard" ratings were given to a flashcard the algorithm would track back.
- Depending on the level of a flashcard, and the rating given, the level will be adjusted: (lvl7, hard = lv4), (lvl7, challenging = lvl5), (lvl7, easy = lvl7), (Lvl6, hard = lvl3), (lvl6, challenging = lvl4), (lvl6, easy = lvl7), (lvl5, hard = lvl3), (lvl5, challenging = lvl4), (lvl5, easy = lvl6), (lvl4, hard = lvl3), (lvl4, challenging = lvl4), (lvl4, easy = lvl5), (lvl3, hard = lvl2), (lvl3, challenging = lvl2), (lvl3, easy = lvl4), (lvl2 , easy/challenging = lvl2), (lvl2, easy = lvl3), (lvl1, easy/challenging/hard = lvl2(level 2 means they are visited everyday))
- Furthermore, the ratings given by the user will be averaged every time. With easy having the value "1.00", challenging "2.00", and hard "3.00". The higher the value means the harder the user found the flashcard.
- New flashcard will be presented based on the user's preference on how much they are willing to work daily.
- The last day that revision is possible will be one day before the user input date of the exam, this means every flashcard must be seen at least once, 32 days before the exam is taken.

- With the time available, and the number of total flashcards on level1 (first ever time a user logs in), the workload will be divided. First: the days available will be calculated: input user date – (the first login date + 32) (t), then the amount of flashcard available (N), will be timed by a coefficient "(x)" that gets smaller as the number of total flashcard increases (need to figure this out) and finally is divided by the number of day to calculate the number of cards that need to be seen on a day to possibly see every card 32 days before the SJT exam.

$$N \times x\ tN \times x\ t$$ **(SCRAPED)**

Workings

The algorithm will be time based; the output of the equations will be the day a flashcard is revised after the initial time it was seen. For this we imagine two separate containers, each given day will have its own unique set of containers. Inside, there will be flashcard material that needs to be revised or seen for the first time. The level1 flashcards, the new materials, will be uniquely inside a container aimed at new flashcards. Every other level except ,level 7, flashcards, will be inside the revision container with respect to the algorithm. at the end of each month (day 31,30 or 28) there will be a third container which holds level 7 flashcards. These are mastered flashcards that are visited only once a month and are ranked based on average rating the user has given them.

Assuming a user is active daily:

A flashcard is allowed to enter a container when the user has seen it and rated it. Meaning a given flashcard can only be uniquely in exactly one container at a given time, priority is given to the revision container in an event where the mastery container is present, and a flashcard has the possibility of being promoted to a level 7. A flashcard will move based on the user's rating confidence and level it is at.

Algo 2.0:

- We are ditching the idea of levels and going instead with the idea of ratings only
- The ratings will be out of 10 instead of 3.
- A given flashcard will be viewed at most 4 times.
- We will have 2 containers still one will be regarded as new material while the other will be dedicated t revision
- The revision algo is scraped instead we will use context mapping at let the user revise what they want to revise
- We can add a progression system (like Duolingo) where we rank the user based on "Experience" rather than mastery… basically the more they do it, the higher their XP.
- When revising a given topic, the cards will be sorted and presented based on average of the ratings the user gave them, hardest appearing first and easiest at last.
- For new materials, we can keep the context mapping with XP system but set them to "zero" as the user is yet to use them.

- For Spaced repetition, we will only use it as a reminder, meaning on a given day a notification will be sent to the user telling them to revise a previously visited subject, we use the above algo for it ($\sum_0^n 2^n$). Apart from that, it is up to user to revise.
- Quizzes are another part of the materials… quizzes will be made of all the material the user has seen. The quiz will use the average ratings the user inputted
- Quizzes will pick questions and flashcards that average between 6-7… meaning if a quiz has 20 questions… the average of all of them will be 6/10…
- Furthermore, the quiz needs to select an equal quantity from all the revised topics… eg: 4 major topics revised by the user = 5 of each topic and all average is 6/10.
- When there is no possibility of equal division, the topic with higher average rating and less XP will take priority to have more questions in the quiz.
- Revision exams will be presented to the user at 3 stages, depending on the inputted exam date.
- One will be halfway through to the exam, one 1 month before, and one a week before. (suggestions not enforced)
- If the user wishes to view the mock exams, they will be available to them regardless.
- In-depth analysis of the mock exams, based on their results can be used to show them where they made the most mistakes.