Problem 4:

My initial solution plan:

- Read in all of the bugs, and student levels

Key considerations:

- should print NO if the hardest bug is too expensive or too hard for my best student
- The more students i have the quicker i can solve all of my bugs
- I should map which best students i can get within the budget.
- Pigeon hole principle with the students i have and the bugs left to decide how many days i can fix everything

Upon implementation, all of my considerations were correct except the last point. Turns out it was not possible find the optimal days in constant time and i had to binary search the possible days within the range `[1, number_of_bugs]` .

I also initially tried to find all of the possible students in one go and then try the binary search, turns out it wouldve made more sense to use binary search first, and with the max number of days assign the chunk of bugs to students and keep trying like that. I think this mistake could have been prevented with better planning.