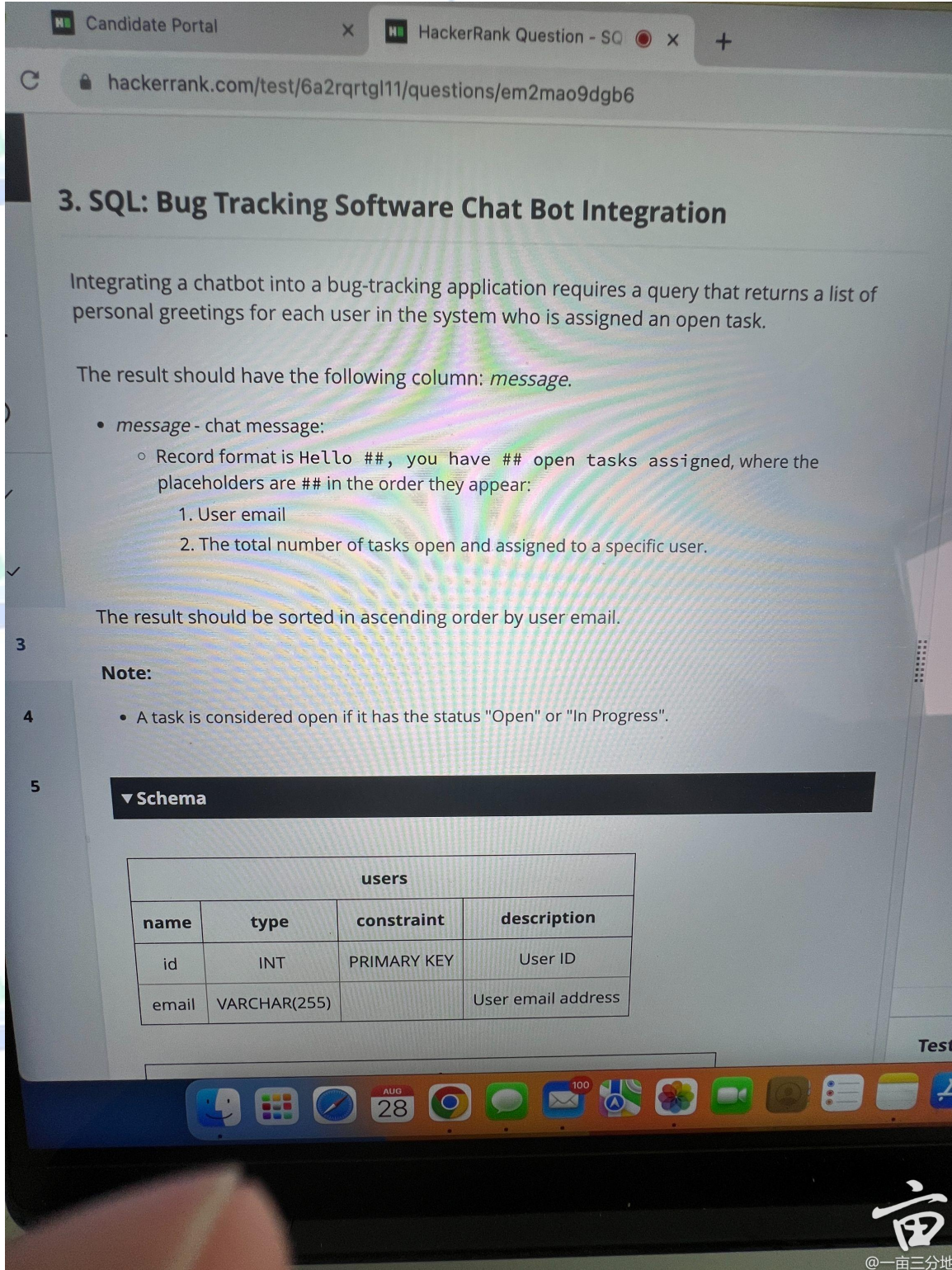


Tiktok - Software Engineer Intern - OA

总共五道题：2 道选择题 + 1 道 SQL + 2 道算法题



The screenshot shows a HackerRank question page. The title is "3. SQL: Bug Tracking Software Chat Bot Integration". The problem description states: "Integrating a chatbot into a bug-tracking application requires a query that returns a list of personal greetings for each user in the system who is assigned an open task. The result should have the following column: *message*." The result format is specified as "Hello ##, you have ## open tasks assigned, where the placeholders are ## in the order they appear: 1. User email 2. The total number of tasks open and assigned to a specific user." The result should be sorted in ascending order by user email. A note states: "A task is considered open if it has the status 'Open' or 'In Progress'." A schema section shows a table named "users" with columns: name, type, constraint, and description. The table structure is as follows:

name	type	constraint	description
id	INT	PRIMARY KEY	User ID
email	VARCHAR(255)		User email address

The screenshot also shows a macOS dock at the bottom with various application icons. The watermark "西游教育 SeeU International Inc." is visible across the image.

1h 30m left

4. Maximizing Task Priority

ByteDance is working on a new algorithm that operates on a special CPU that can handle n types of tasks, and the priority of the i^{th} task is represented by the array $priority[i]$. The CPU will run for y seconds and you can choose any task to perform which follows the given conditions:

- In each second, you user can choose to perform any available task from the n types (if there are any tasks to perform).
- It is allowed to perform multiple tasks of the same type within the given time frame, but there is a time constraint that there must be at least x seconds between any two consecutive tasks of the same type.
- It is not allowed to perform more than one task at any second.

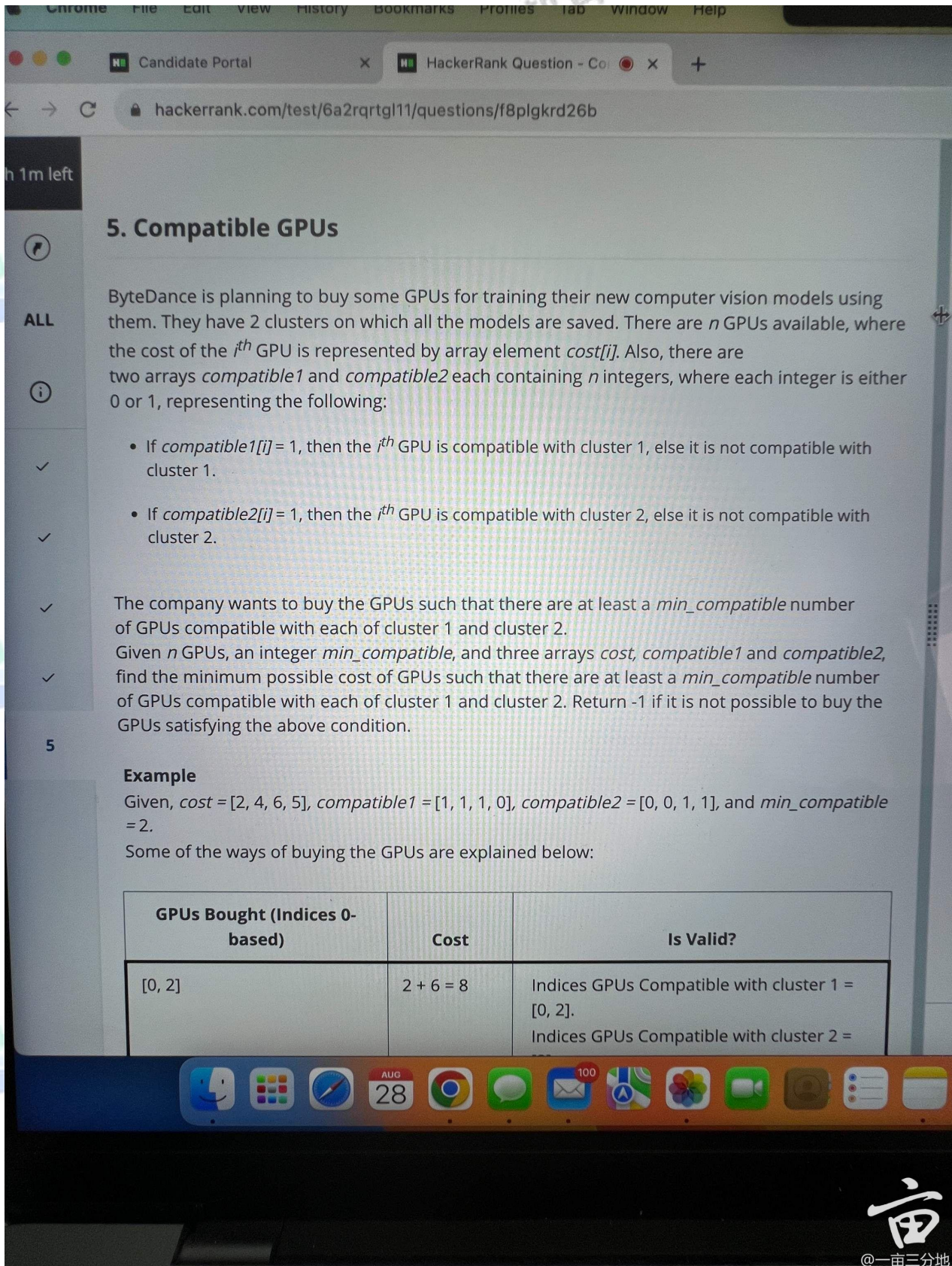
Given n tasks, an array $priority$, and an integer x and y , find the maximum possible sum of the priority of tasks performed during the y seconds.

Example
Given, $n = 3$, $priority = [3, 1, 2]$, $x = 5$, and $y = 7$.

Seconds	Task Performed	Sum of priority
1.	Perform a task of the 1 st type	3
2.	Perform a task of the 3 rd type	3 + 2 = 5
3.	Perform a task of the 2 nd type	3 + 2 + 1 = 6
4.	No task performed	3 + 2 + 1 = 6
5.	No task performed	3 + 2 + 1 = 6

Test Result

@一亩三分地



h 1m left

5. Compatible GPUs

ByteDance is planning to buy some GPUs for training their new computer vision models using them. They have 2 clusters on which all the models are saved. There are n GPUs available, where the cost of the i^{th} GPU is represented by array element $cost[i]$. Also, there are two arrays $compatible1$ and $compatible2$ each containing n integers, where each integer is either 0 or 1, representing the following:

- If $compatible1[i] = 1$, then the i^{th} GPU is compatible with cluster 1, else it is not compatible with cluster 1.
- If $compatible2[i] = 1$, then the i^{th} GPU is compatible with cluster 2, else it is not compatible with cluster 2.

The company wants to buy the GPUs such that there are at least a $min_compatible$ number of GPUs compatible with each of cluster 1 and cluster 2.


Given n GPUs, an integer $min_compatible$, and three arrays $cost$, $compatible1$ and $compatible2$, find the minimum possible cost of GPUs such that there are at least a $min_compatible$ number of GPUs compatible with each of cluster 1 and cluster 2. Return -1 if it is not possible to buy the GPUs satisfying the above condition.


Example

Given, $cost = [2, 4, 6, 5]$, $compatible1 = [1, 1, 1, 0]$, $compatible2 = [0, 0, 1, 1]$, and $min_compatible = 2$.

Some of the ways of buying the GPUs are explained below:

GPUs Bought (Indices 0-based)	Cost	Is Valid?
[0, 2]	$2 + 6 = 8$	Indices GPUs Compatible with cluster 1 = [0, 2]. Indices GPUs Compatible with cluster 2 =




 @一亩三分地

Example 是 $return 2+5+6$