Answer1:

a: The variable top can have a race condition

b: One way to fix the race condition is to use a locking mechanism, such as a mutex or semaphore, to ensure that only one thread can access or modify the shared data at a time.

Answer2:

The trade-off in the readers-writers problem is between favoring multiple readers for high throughput versus potentially starving the writer(s) of access. To prevent starvation, a timestamp-based approach can be used to ensure fairness, where the longest waiting process is given priority. Additionally, a fairer solution can be achieved by creating a queue for writers and allowing them to have exclusive access to the semaphore after the currently reading readers are done.

Answer3:

a: available_resources
b: The race condition occurs in the decrease_count().