J244 - Passing Rate

Daniel Hsieh {QwertyPi}

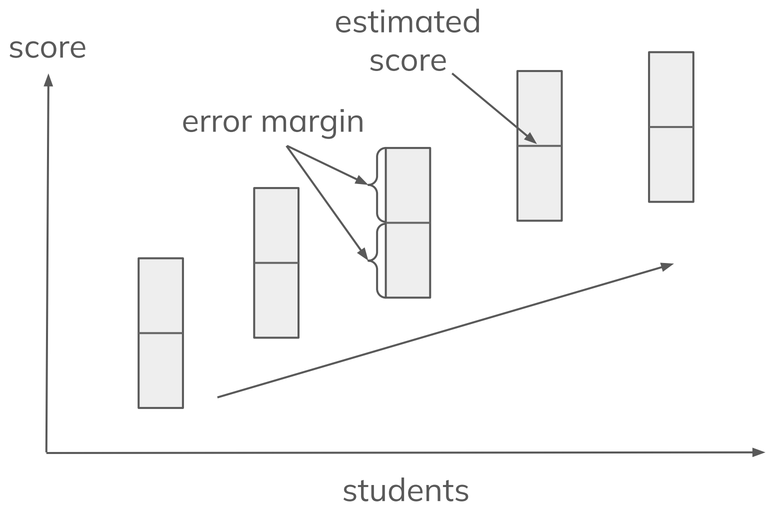
2024-02-17

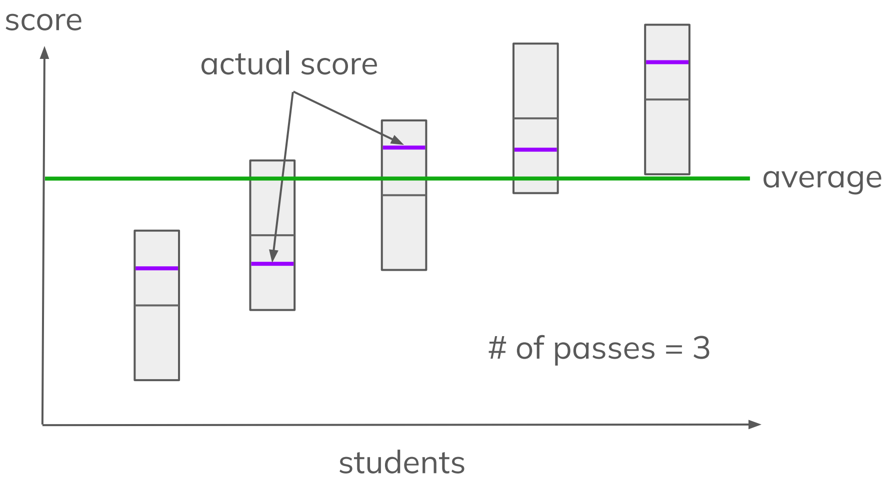
Background

Problem Idea and Preparation by QwertyPi

Problem Restatement

Given estimated scores of N students. Your task is to maximise the number of students obtaining a score not less than the average, by cleverly choosing an integer score for each of the students for Q different error margins.





Subtasks

For all cases:

$1 \le N, Q \le 10^5$

$1 \le M \le 10^9$

$0 \le A_1 \le A_2 \le \cdots \le A_N \le M$

$0 \le D_1 < D_2 < \cdots < D_Q \le \min(A_1, M - A_N)$

both sorted

	Points	Constraints
1	9	$N = 3$ $1 \le Q \le 50$ $M = 100$
2	13	$1 \le N \le 10$ $1 \le Q \le 50$ $M = 100$
3	18	$1 \le N \le 500$ $1 \le Q \le 50$ $M = 100$
4	24	$1 \le N, Q \le 500$
5	15	$1 \le N, Q \le 5000$
6	21	No additional constraints

Statistics

```
  0 points   45 + 20 + 11 + 3 = 79
  9 points    1 +  3 +  4 + 1 =  9
 79 points    0 +  0 +  0 + 2 =  2
100 points    0 +  0 +  0 + 2 =  2
```

First solved by **s20297** at **1:05:59**

Last solved by **s20251** at **2:13:02**

Subtask 1

Subtask 1 (9%): $N = 3$, $1 \le Q \le 50$, $M = 100$

- Since N and M are small, we may enumerate all the possible cases.
- We may use three nested for-loops to do so.
- Be careful not to take the floored average!

Time Complexity: $O(M^3Q)$

Expected Score: 9

```
x1 >= (x1 + x2 + x3) / 3             WA
x1 >= (double) (x1 + x2 + x3) / 3    Accepted
x1 * 3 >= x1 + x2 + x3               Accepted
```

Subtask 2

Subtask 2 (13%): $1 \le N \le 10$, $1 \le Q \le 50$, $M = 100$

- If we do it naively, it will now need $O(M^N)$ operations for a single query.
- It is hard to gain any useful information as when we change one student's score, the average score also changes...
- Can we somehow 'fix' the average score?

Subtask 2

- We introduce a fixed **cutting score** - a student passes iff

  student score ≥ cutting score

- Now, as long as

  cutting score ≥ average score

Then students who passes the cutting score also passes the average score.

Subtask 2

- Instead of considering average scores, we consider **all situations** of

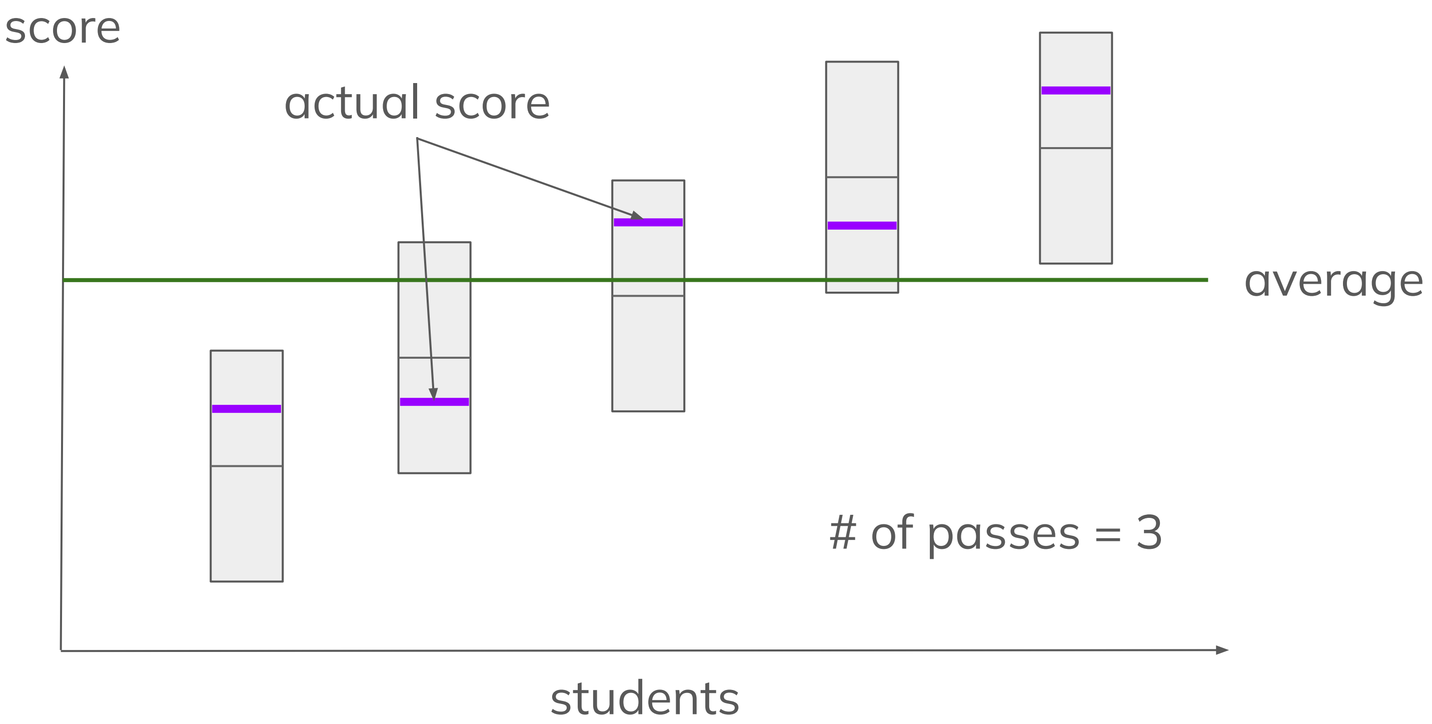
  (score list, cutting score)

  which are **valid**: i.e. the **cutting score ≥ average score**.

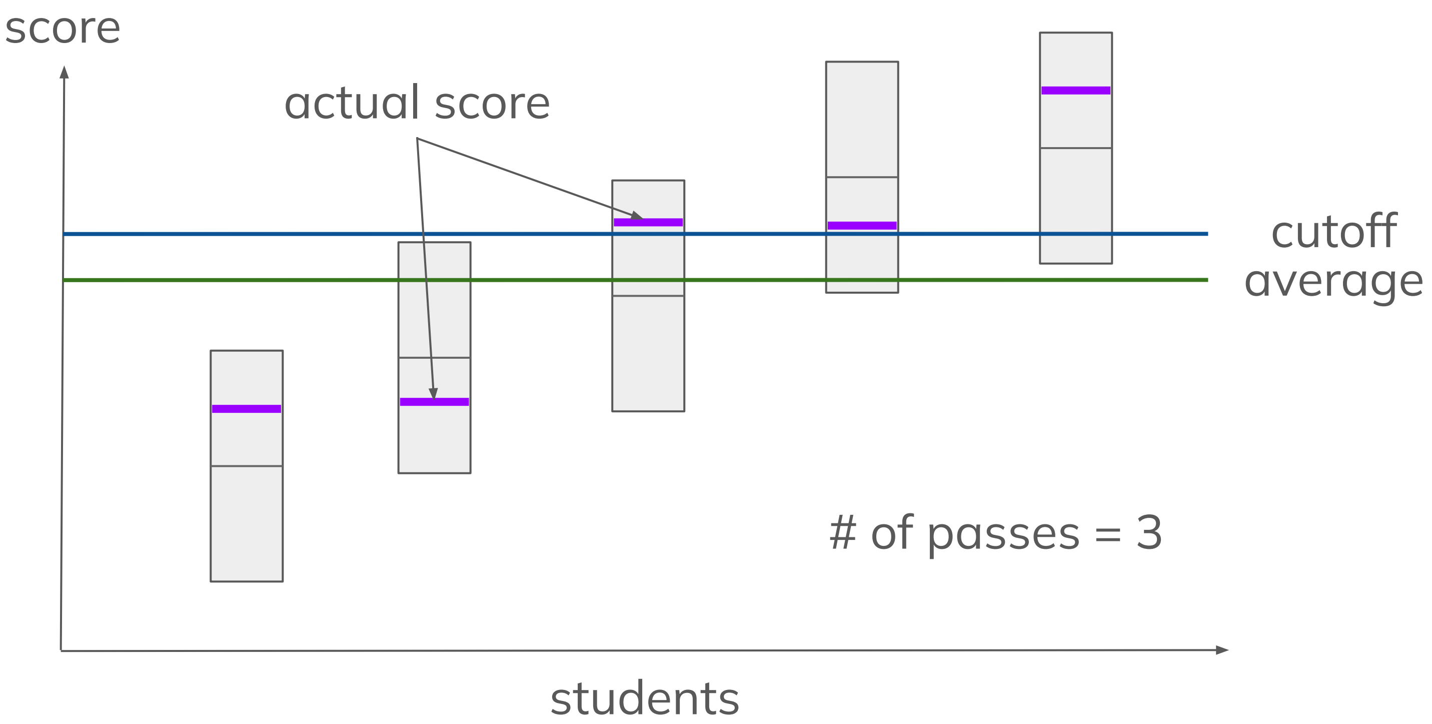
- The problem now is: have we covered all the possibilities?
- Yes, as we may simply take the cutting score = average score!

- More situations are considered - but the answer does not change!

Subtask 2



Subtask 2



Subtask 2

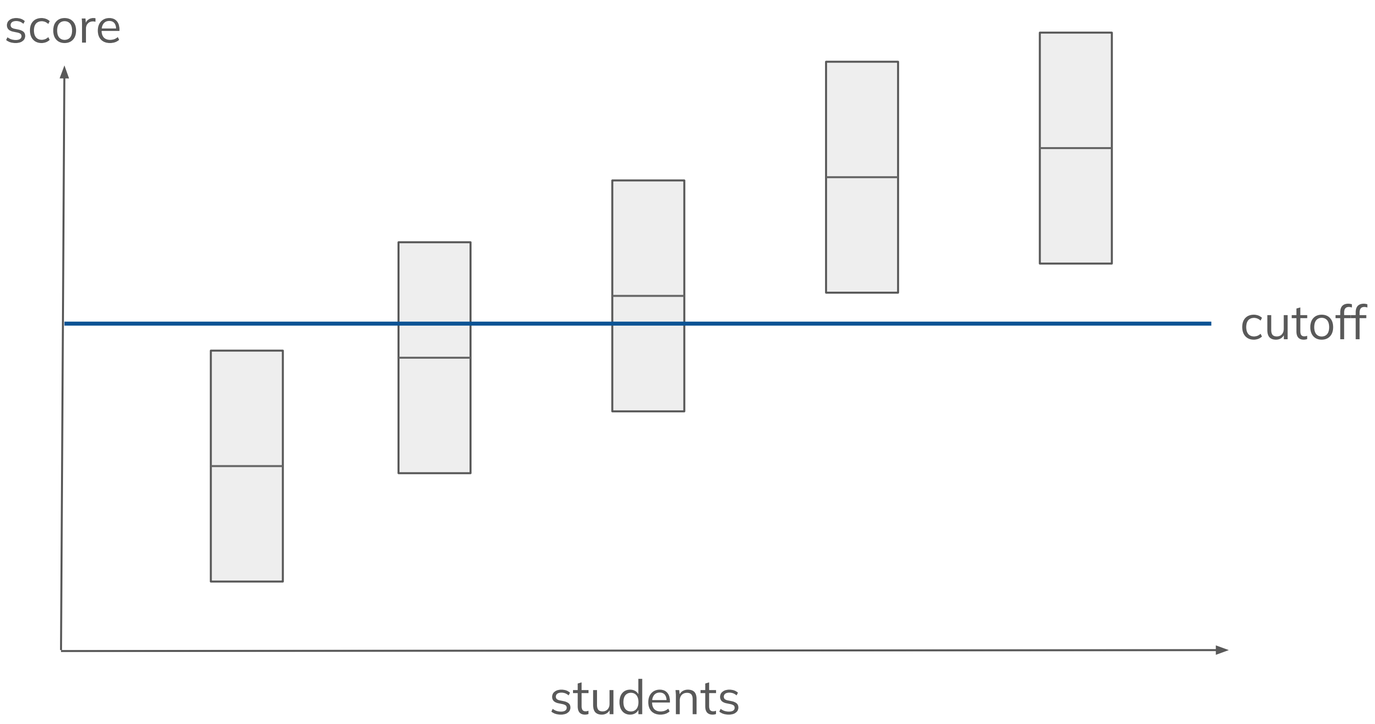
Observation 1. It suffices to consider only **integral cutting scores**.

- Notice that the score of the students must be integer.
- Therefore, we can take **cutting score = ceil(average score)**.

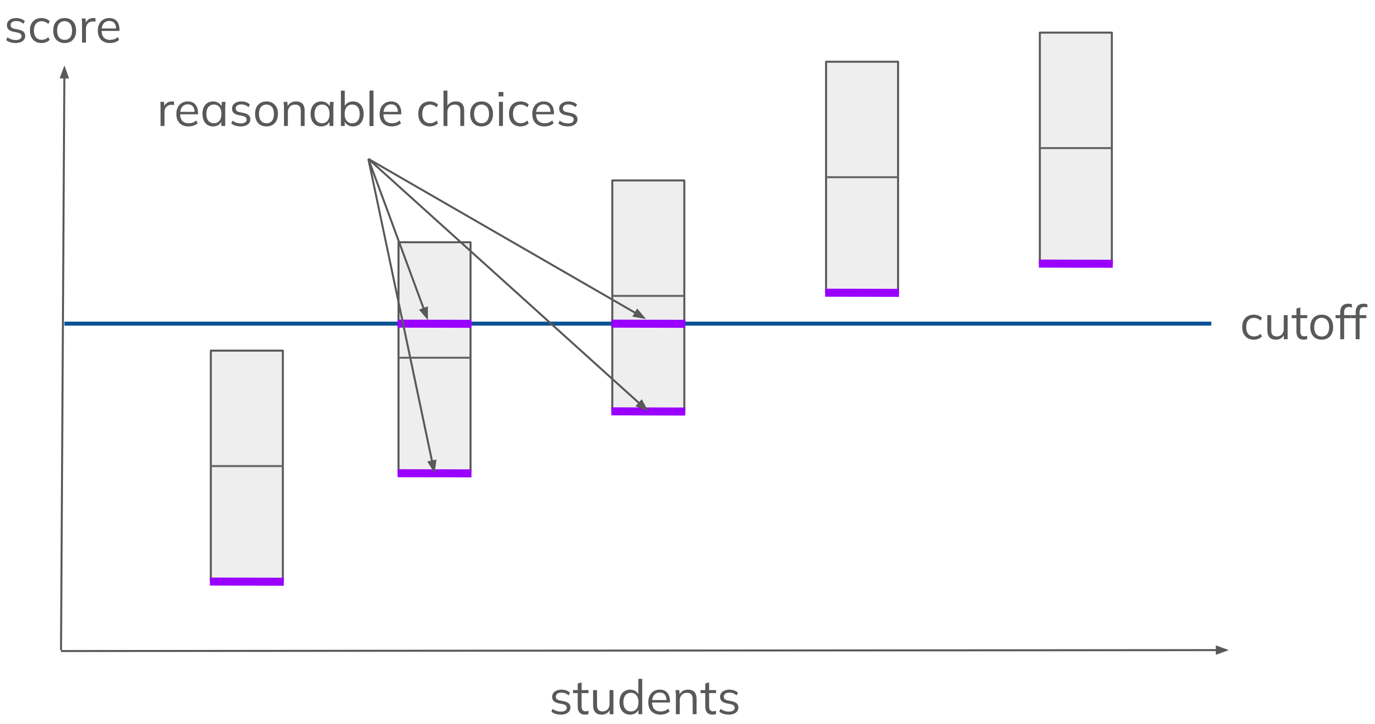
Observation 2. For a **fixed cutting score C**, for each student with a possible score of between L and R, the student's actual score should be **either L or C**.

- Our target is to make **cutting score ≥ average score**.
- As the **cutting score is fixed**, we want to **minimise the average score without affecting pass / fail**.
- Therefore, as long as we do not meet any 'boundary', we can repeatedly reduce the scores of the students.

Subtask 2



Subtask 2



Subtask 2

- For this subtask, we can consider all possible cutting scores and choose either L or C for each student independently.

- You may want to use bitwise number to represent the the score choices of the students.

Time Complexity: $O(M2^NQ)$
Expected Score: 22

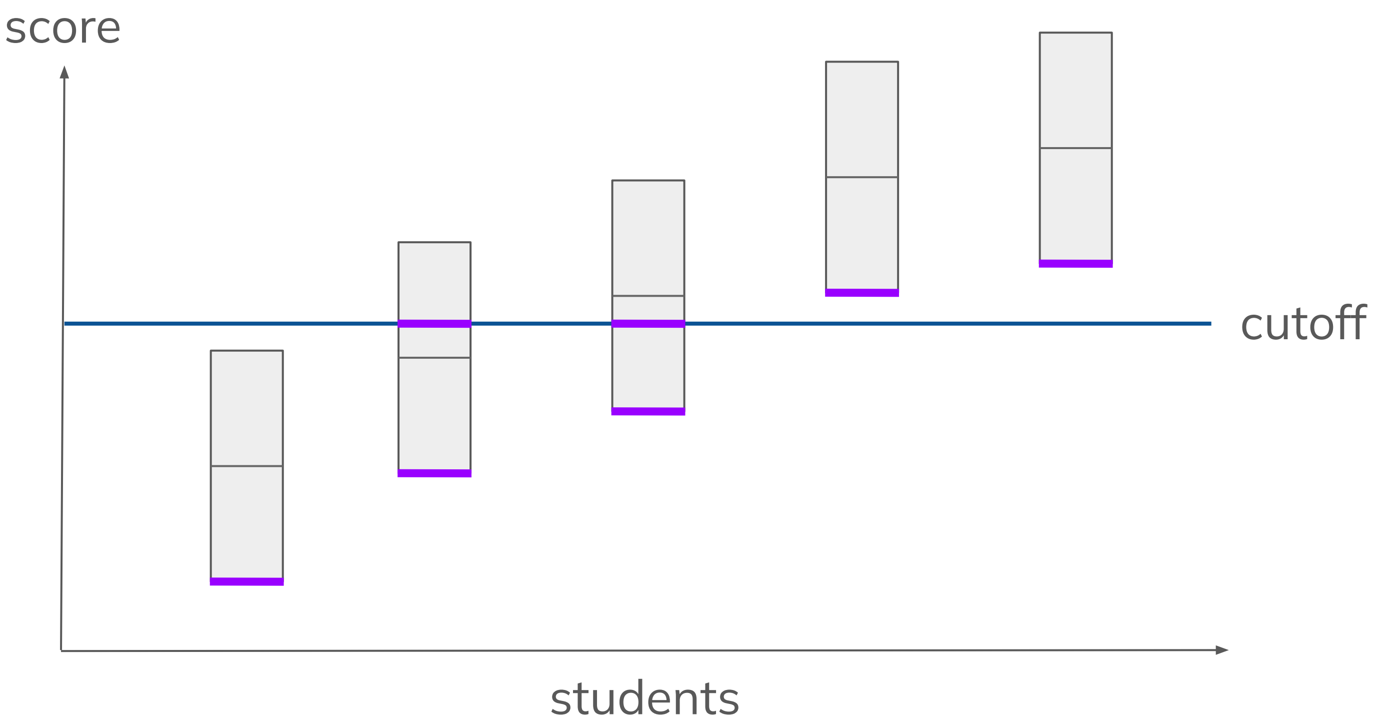
Subtask 3

Subtask 3 (18%): $1 \le N \le 500$, $1 \le Q \le 50$, $M = 100$

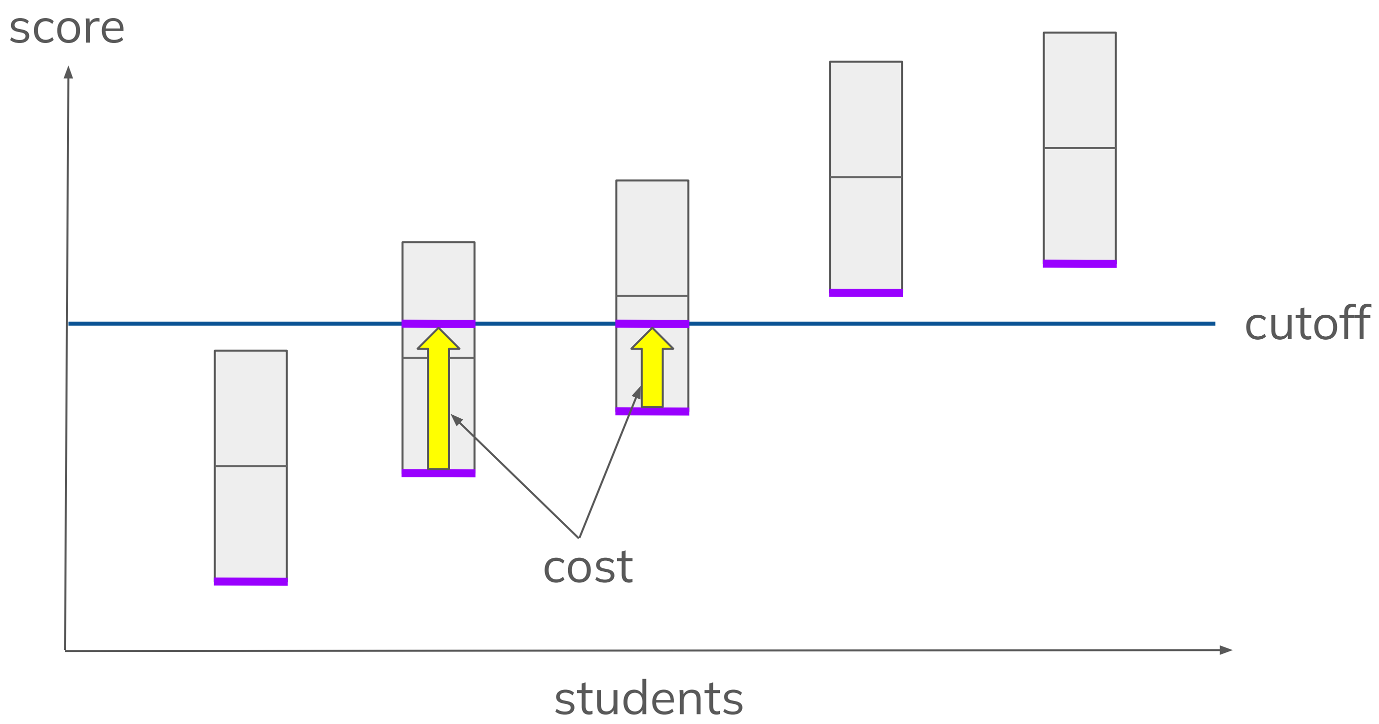
- Clearly, we cannot be too brute force from now on, and we need some more intuitive ideas!

- Question: When would we choose a student with score of C instead of L?
  - The student does not pass with score L, but passes with score C.
  - This gives us one more pass, in cost of that the total score will increase by C - L.

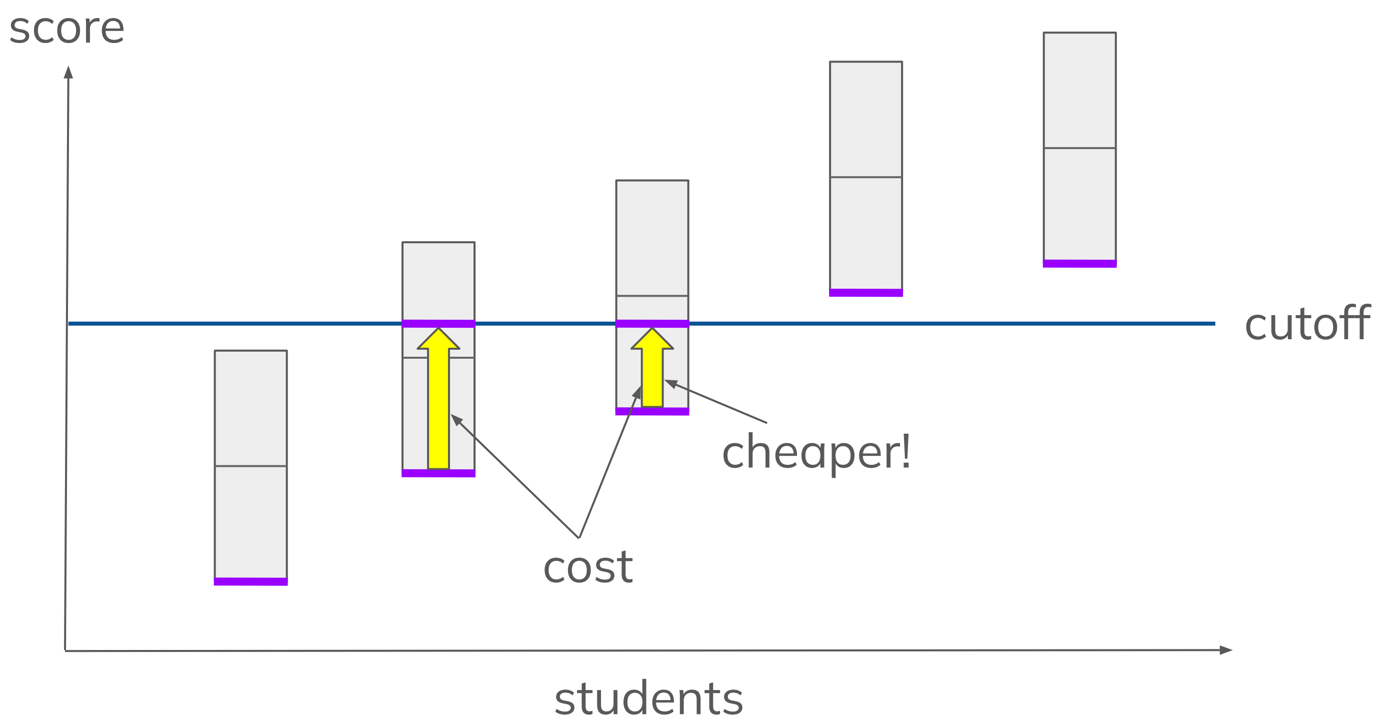
Subtask 3



Subtask 3



Subtask 3



Subtask 3

- Therefore, we may use greedily increase student's score from lower bound L to cutting score C, as long as the average score does not exceed the cutting score, whoever with the minimum score difference.
- Notice that the estimated score are sorted, so actually the student with smaller score difference is always located on the right. This gives us a relatively easier way to greedy.

Time Complexity: O(NMQ)
Expected Score: 40

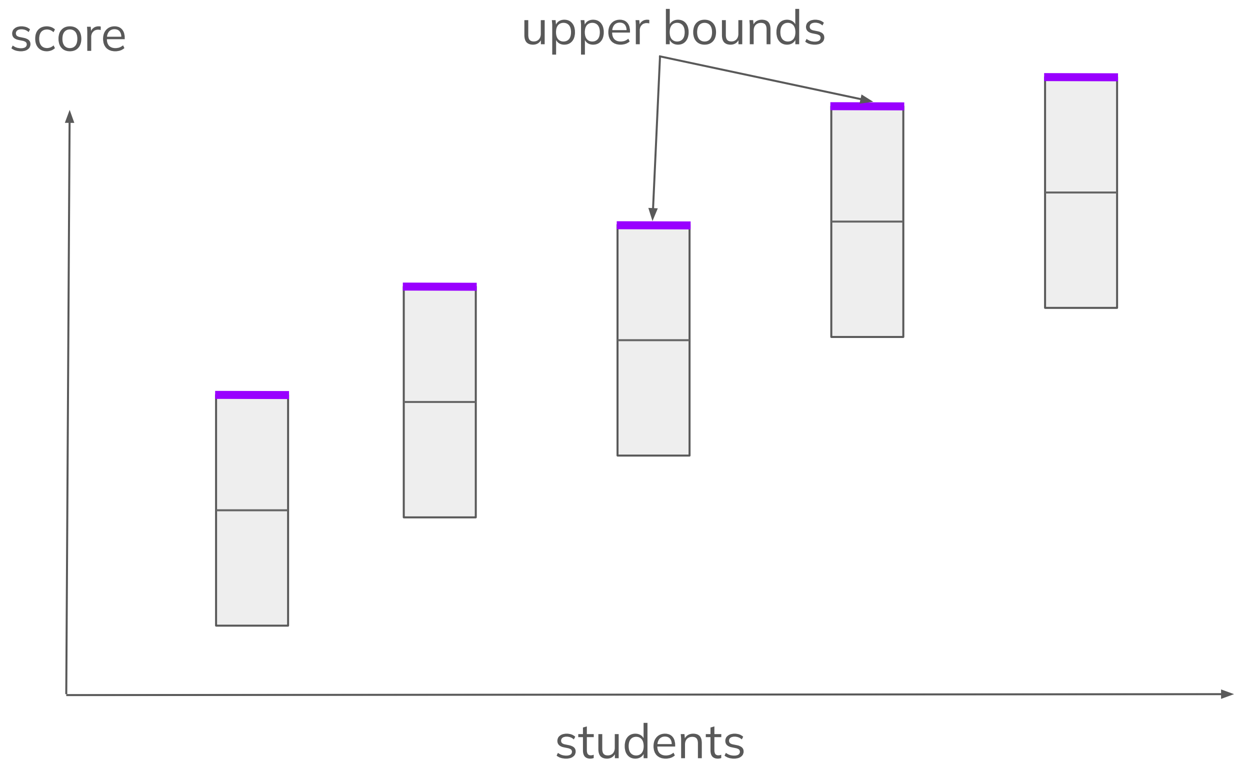
Subtask 4

Subtask 4 (24%): $1 \le N, Q \le 500$

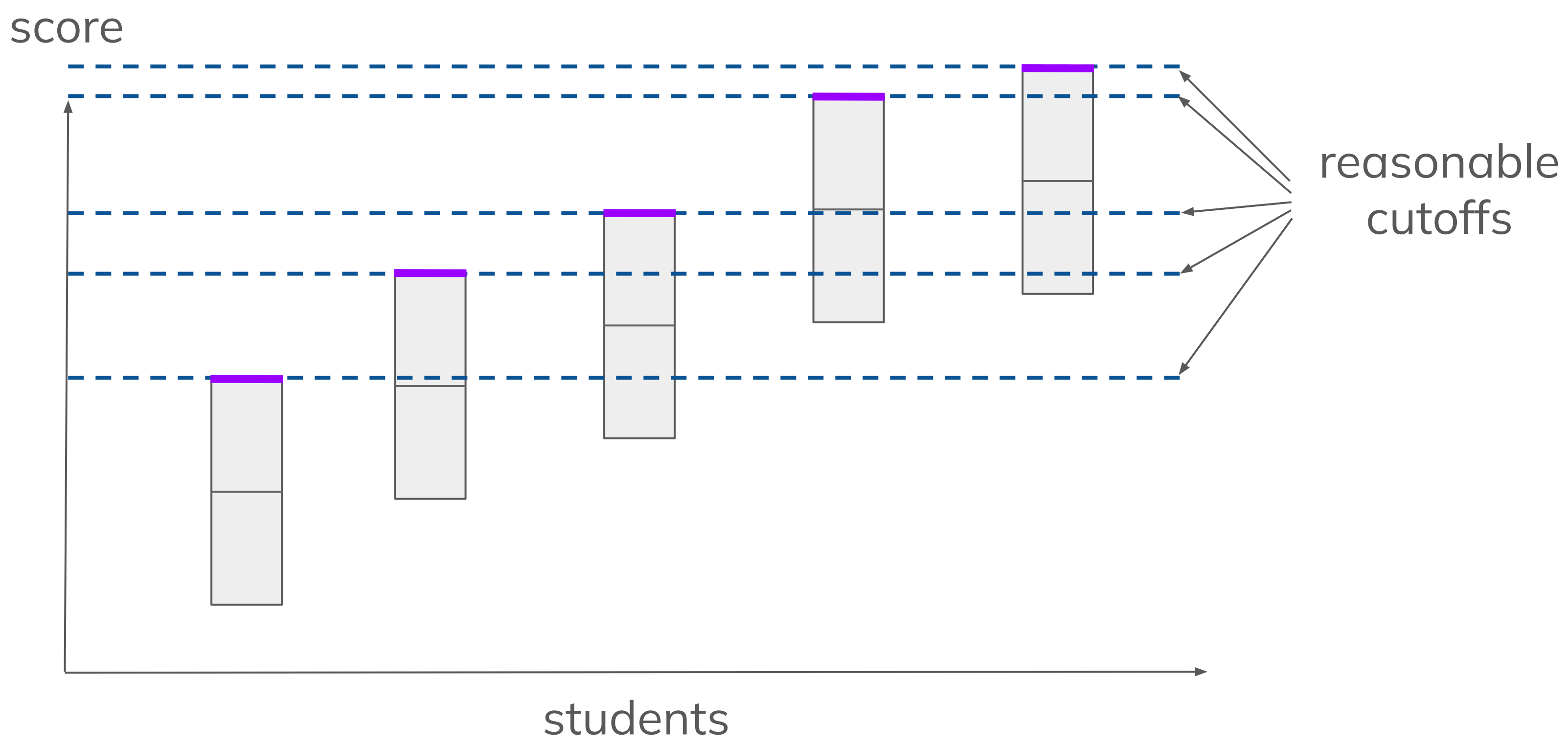
Observation 3. It suffices to consider **cutting scores being upper bound score** of some student.

- Suppose otherwise.
- We can increase the score of every student at the cutoff by 1.
- Meanwhile, the average score only increases by ≤ 1. (Why?)
- Therefore, there must be equal or more passing students.

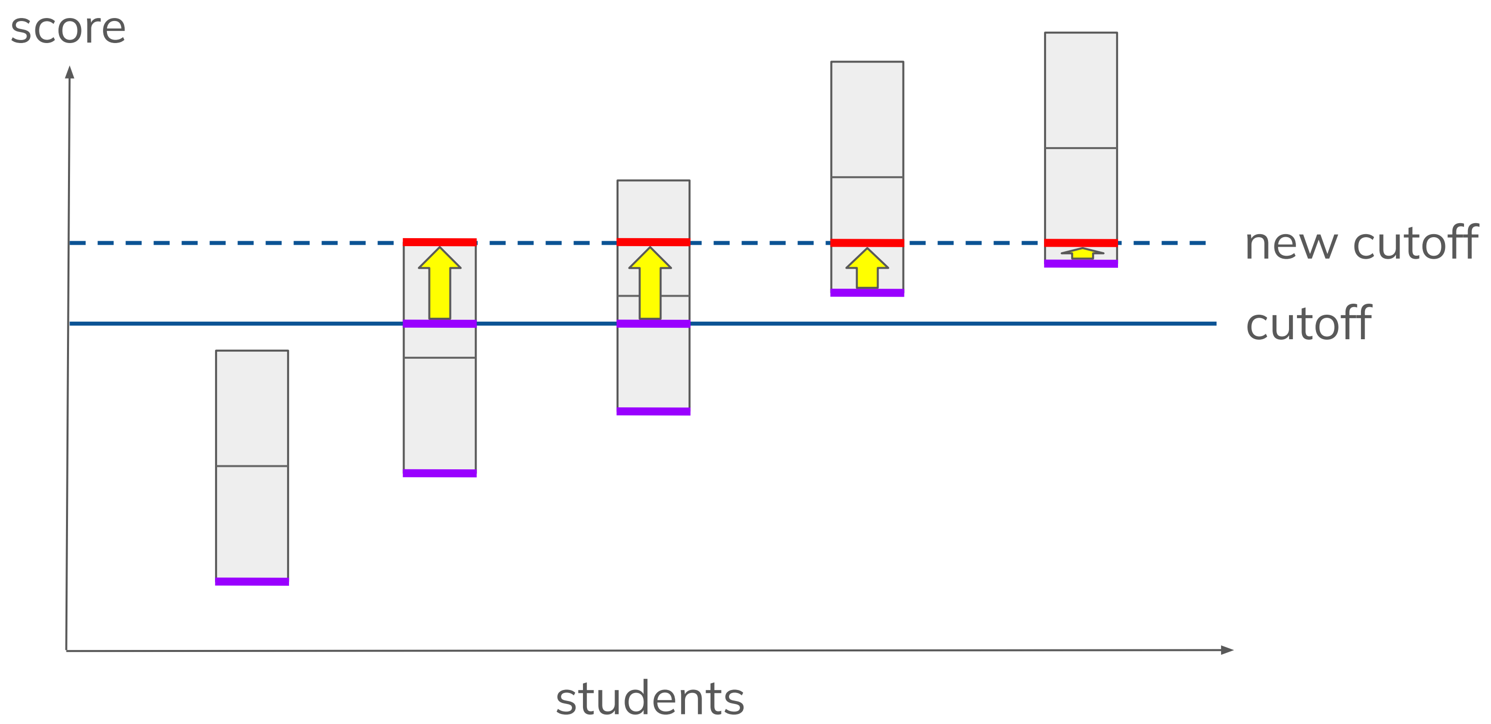
Subtask 4



Subtask 4



Subtask 4



Subtask 4

- Therefore, we can apply the solution for subtask 3 directly while considering at most N cutting scores.

Time Complexity: $O(N^2Q)$

Expected Score: 64

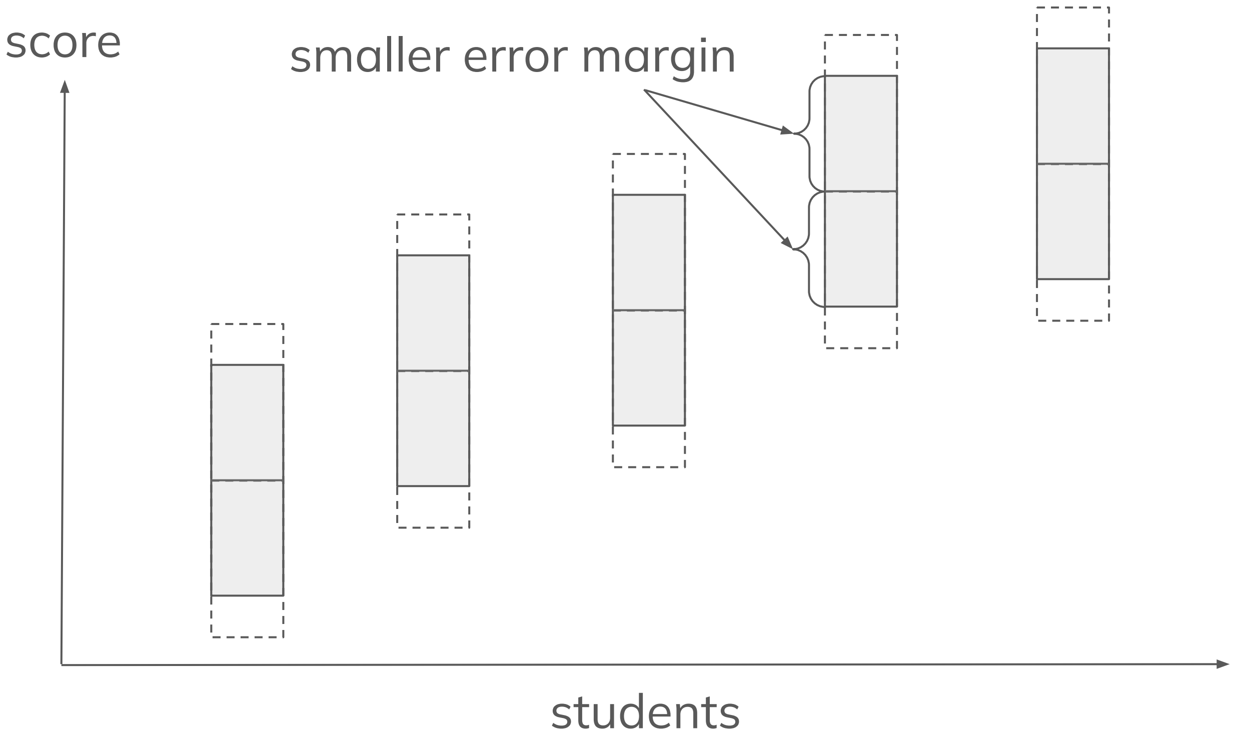
Subtask 5

Subtask 5 (15%): $1 \le N, Q \le 5000$

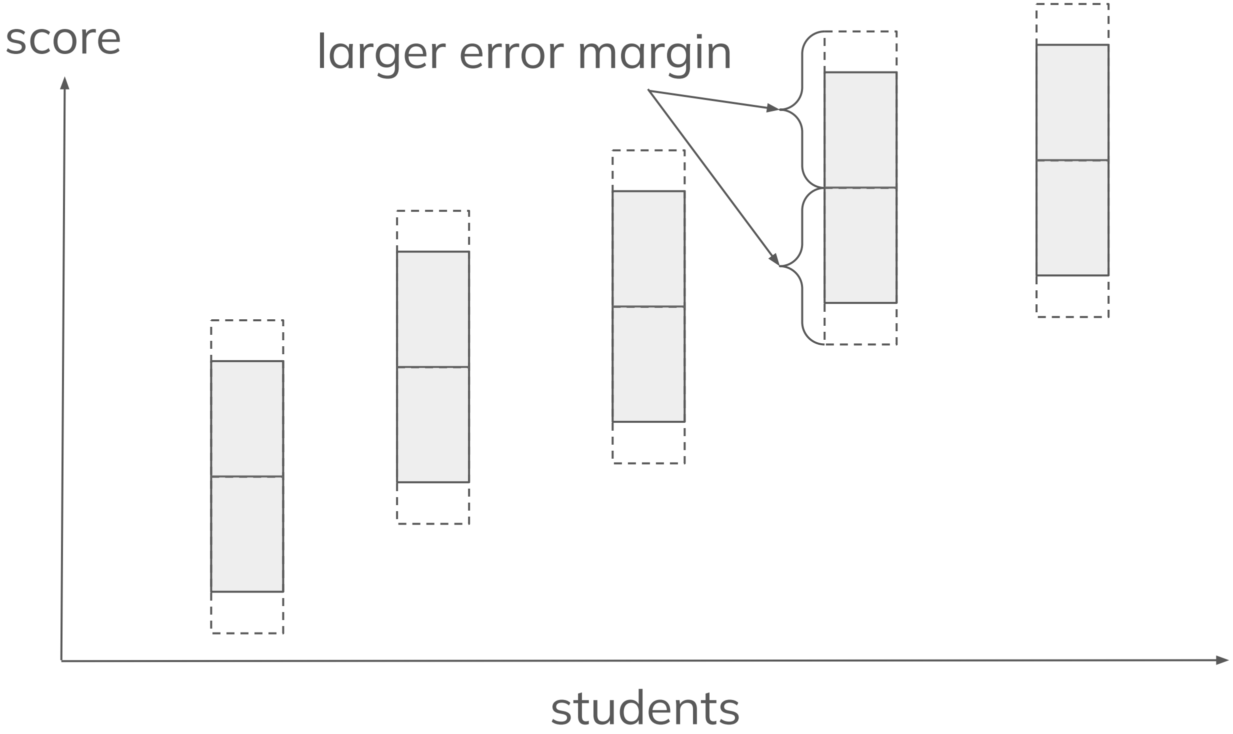
Observation 4. The **larger the error margin**, the **max no. of student passes increases or remains unchanged**.

- Indeed, any score list attainable with a smaller error margin can be attained with a larger error margin!

Subtask 5



Subtask 5



Subtask 5

- As a result, you can ‘preserve’ the answer for previous queries (with smaller error margin), and increase the answer whenever possible.

- Suppose that we implemented the followings:
  `check(D, K)` Is it possible to have K students pass with error margin D?

- Possible pseudocode:

```
ans = 0
for query with error margin D:
    while ans < N and check(D, ans + 1):
        ans++
    output ans
```

Time Complexity: O((N + Q) N)
Expected Score: 79

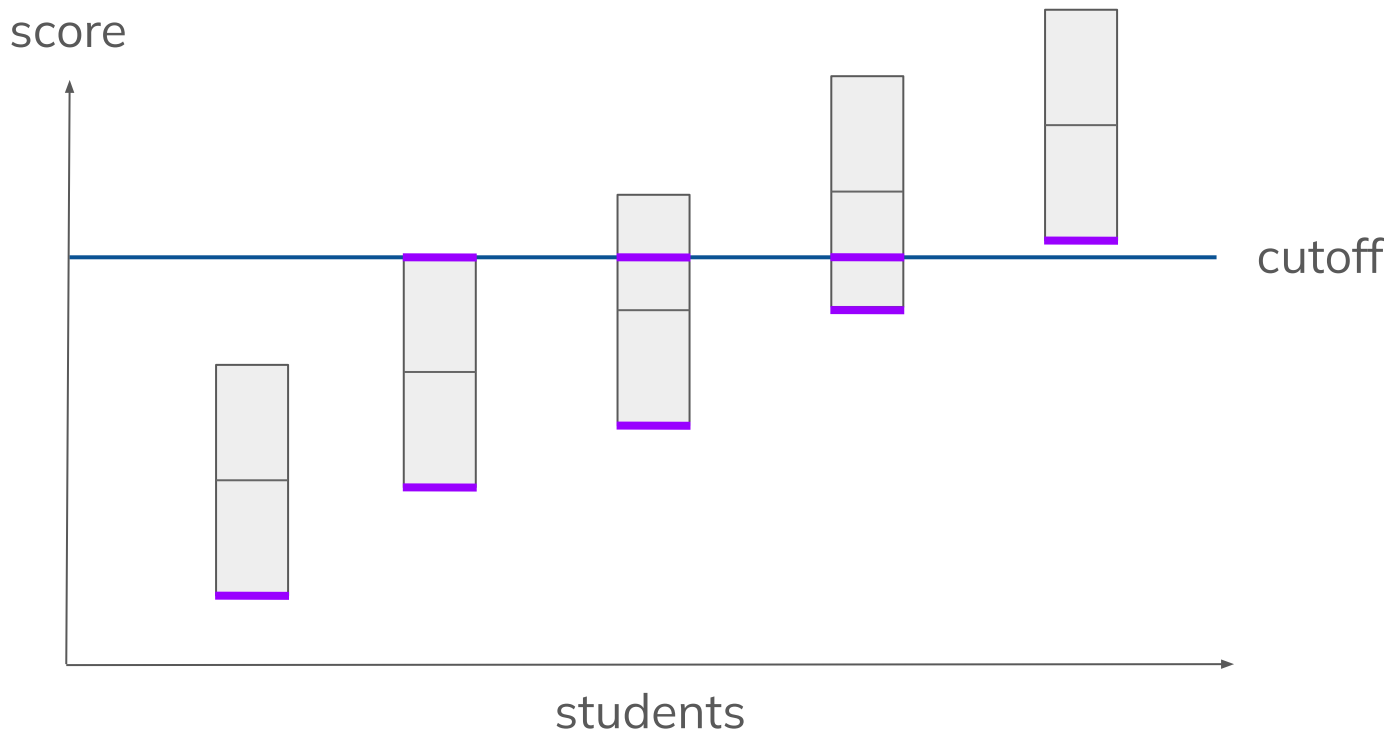
Subtask 5

- Alternatively, you can also use binary search + partial sum to optimise the greedy algorithm in subtask 3.

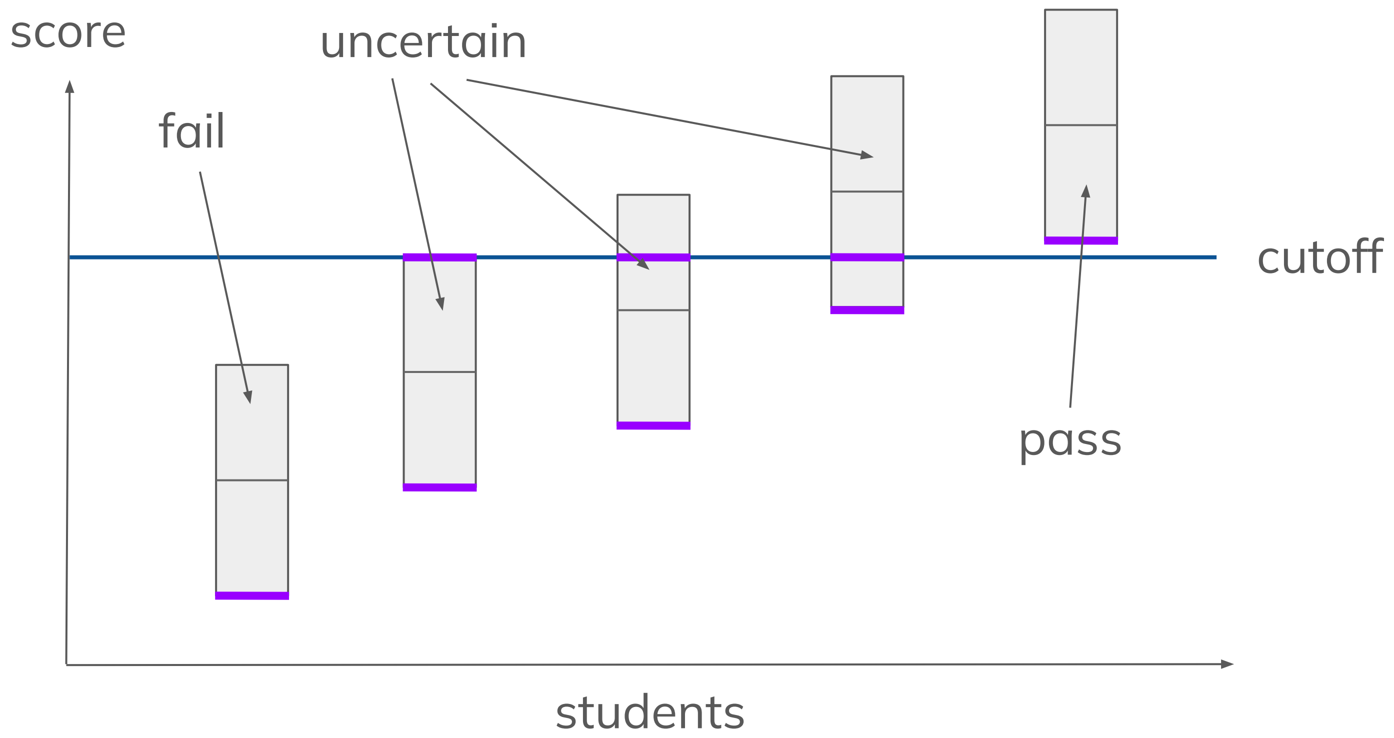
Solution Sketch:

- We may put students in three groups: **fail**, **uncertain** and **pass**.
- You can **binary search on index** for the intervals.

Subtask 5



Subtask 5



Subtask 5

- Recall Observation 2:

For a **fixed cutting score C**, for each student with a possible score of between L and R, the student's actual score should be **either L or C**.

- For **fail** or **pass**, we must be choosing L.
- For **uncertain**, we choose C for some on the right (↓cost) and choose L for others.

- Hence, we can **binary search on answer** to see how many Cs can be chosen for **uncertain**.
- When more C is chosen instead of L, the average score is higher, and so the target is to find the exact answer such that the **average score is just below the cutting score**.

Time Complexity: O(NQ log N)
Expected Score: 79

Full Solution

- Simply combine both ideas of subtask 5 (preserve answer + partial sum)!

Time Complexity: $O((N + Q) \log N)$

Expected Score: 100

Alternative Solution

- Without observation 3 (cutting score = upper bound), you can still technically do binary search on the cutting score.
- This adds another log to the solution, but is still completely fine.

Time Complexity: $O((N + Q) \log^2 N)$

Expected Score: 100

Conclusion

- Somehow only four contestants breakthrough subtask 2...
- It is useful to change your ‘perspective’, not to be limited by the problem
    - Most of the time, you can ‘modify’ the problem to something easier
- You may draw simple graphics on your rough work paper
    - Ensure you can understand the problem / to observe something