Problem Statement:

A particular school offers cash rewards to children with good attendance and punctuality. If they are absent for three consecutive days or late on more than one occasion then they forfeit their prize.

During an n-day period a trinary string is formed for each child consisting of L's (late), O's (on time), and A's (absent).

Although there are eighty-one trinary strings for a 4-day period that can be formed, exactly forty-three strings would lead to a prize:

OOOO OOOA OOOL OOAO OOAA OOAL OOLO OOLA OAOO OAOA
OAOL OAAO OAAL OALO OALA OLOO OLOA OLAO OLAA AOOO AOOA AOOL AOAO AOAA AOAL AOLO AOLA AAOO AAOA AAOL AALO AALA ALOO ALOA ALAO ALAA LOOO LOOA LOAO LOAA LAOO LAOA LAAO

How many "prize" strings exist over a 30-day period?

Solution:

I was not familiar with the concept of ternary strings – hence to gather some idea I went through some example on stack exchange and reddit: References below:

https://math.stackexchange.com/questions/1524771/combinatorics-ternary-strings
https://www.reddit.com/r/learnmath/comments/2glzv5/combinatorics_ternary_string_questions/

Reading through I understand that we can have 3^n possible strings of Os and As and Ls.
Where n = no_of_days;

These strings could be combinations of various sequences,
I came up with these:

Strings with Sequence with consecutive "AA" (OAA)
Strings with no A(OLL) (OOO)
Strings with consecutive "AA" and one L(LAA)(AAL)
Strings with A and L (OAL)
Strings with only L (OOL)

(Not sure if I got all of them correct or missed combinations or completely incorrect ☹)

I see that I have to iterate over days ,number_of_absent_days, num_of_late_days to determine the consecutive absent strings (AA) sequence and for presence of any Late (L's)

I created a multi dimensional array using the below concept:

```
var  DAYS = 31;
//count the days until 'n' days => 30 days and break when days= 31;
```

var ABSENT_ALLOWED = 3 //number of ABSENTs(ab > = 3)
var LATE_ALLOWED = 2 // (late >1)

Using Array fill to fill in all the empty nested arrays with 0 for recursive adding of values while iterating over the n days.

```
var arr = new Array(31).fill(0);

for (var i = 0; i < arr.length; i++) {
  arr[i] = new Array(3).fill(0);
  for (var j = 0; j < arr[i].length; j++) {
   arr[i][j] = new Array(2).fill(0);
  }
}
```

Iterating over ‘n’ days:

for (var i = 1; i <= DAYS; i++) { }// 1st level of iteration
for (var j = 0; j <= ABSENT_ALLOWED; j++) {}// 2nd level of iteration
for (var k = 0; k <= LATE_ALLOWED; k++) {}// 3rd level of iteration

once iterated over I have the sequences stored in “arr”

arr[i][j][k] = total;

Final string is to get the concatenation of:

[Days, A, L] = (D-1,A,L) (on time)

+PS(D-1,A,L-1)(late)

+PS(D-1,A-1,L)(absent)
total += arr[DAYS][j][k];

Solution:
https://jsfiddle.net/wsk6ch0p/

References:
https://math.stackexchange.com/questions/1524771/combinatorics-ternary-strings
https://www.reddit.com/r/learnmath/comments/2glzv5/combinatorics_ternary_string_questions/

HackerRank discussion board to get ideas- I came up with the understanding on Dynamic programming. From there I understood he approach of Top-down-where I analyzed the different combinations and sequences possible in the 30 day period

Time:

12-15 hours