Introduction to Algorithms

Lab1 Athlete Schedule

2022/10/20 -> 2022/11/10

Professor: Hung-Ming Chen

Hank, Li

Outline

- Introduction
- Input
- Output
- Grading
- Submission
- Notice

Introduction

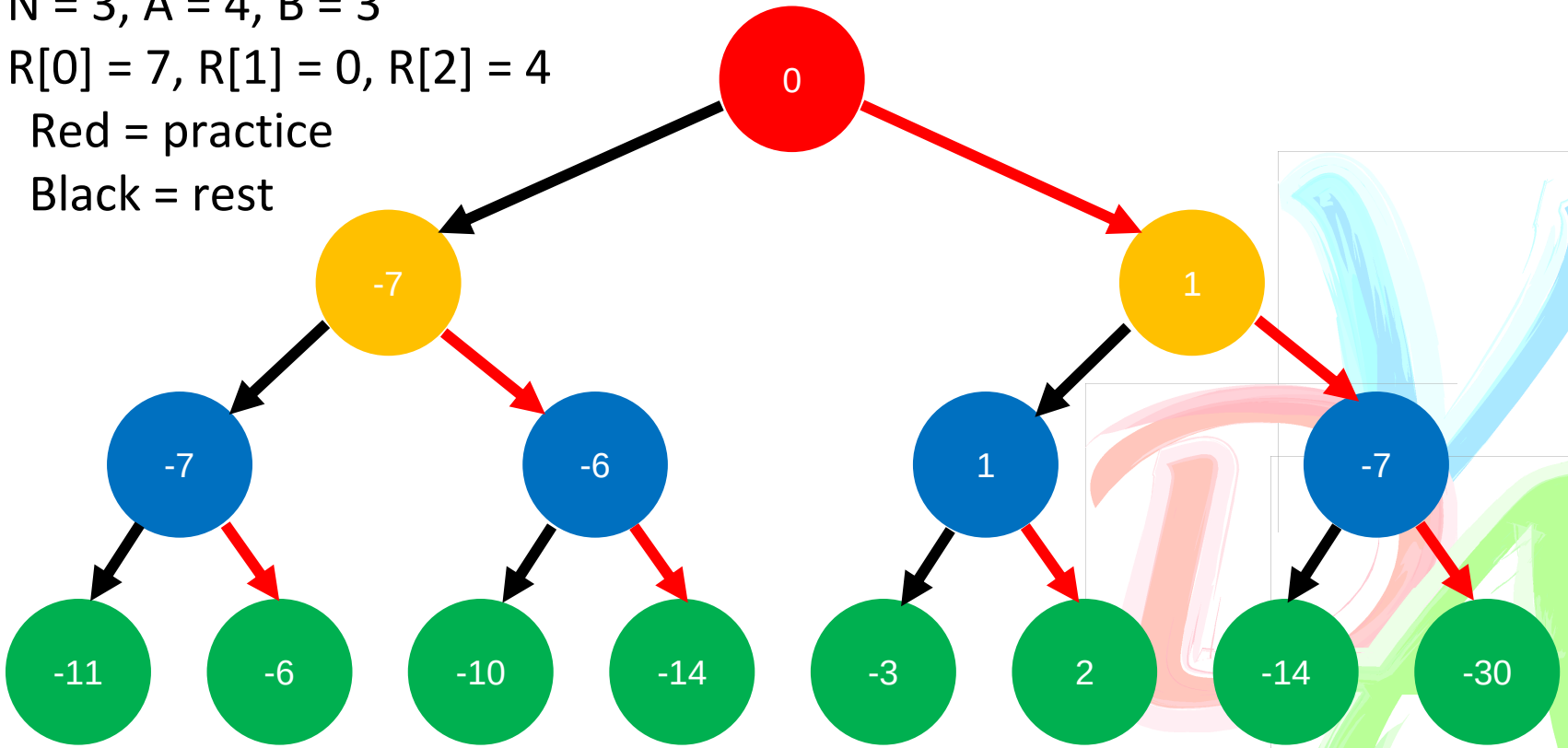
- Objection
  1. Dynamic programming

Introduction

- Introduction
  - You have to manage your rest and practice schedule to get highest performance P in next couple of days.
  - You have N days to prepare for the contest. However, you only have the choice of either rest or practice in a single day.
  - For each day you practice, you will increase some constant number A to the number of performance P .
  - However, without resting for X days in a row, including the day that you have decided to practice, in the end of the day the number of your performance P will be decreased by $X^2 * B$, where B is some constant factor.
  - Additionally, if you have decided to rest in the i'th day, you would start to think that your performance will get worse, and the number of performance P will be decreased by R[i] .

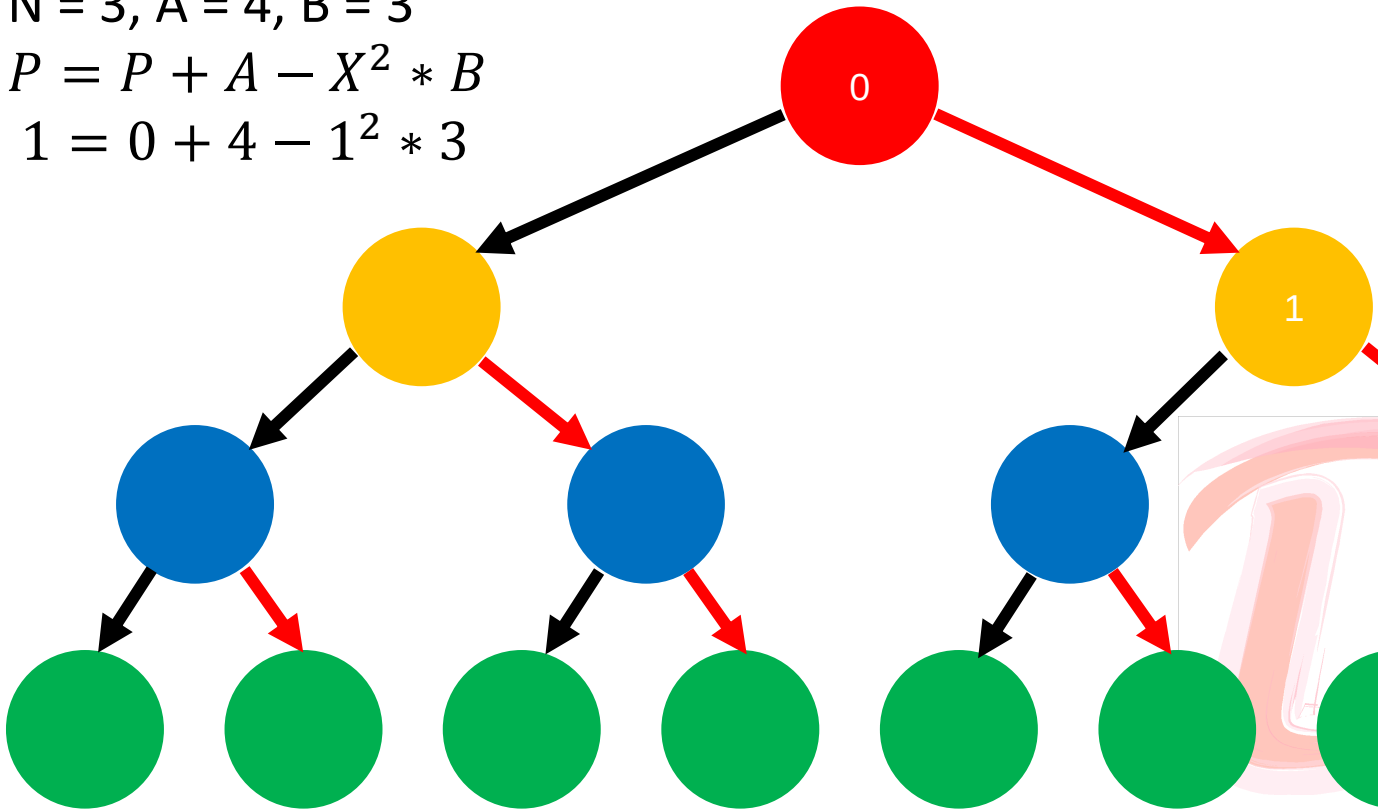
Rules

- Example (input)
  - N = 3, A = 4, B = 3
  - R[0] = 7, R[1] = 0, R[2] = 4
  - ① Red = practice
  - ② Black = rest



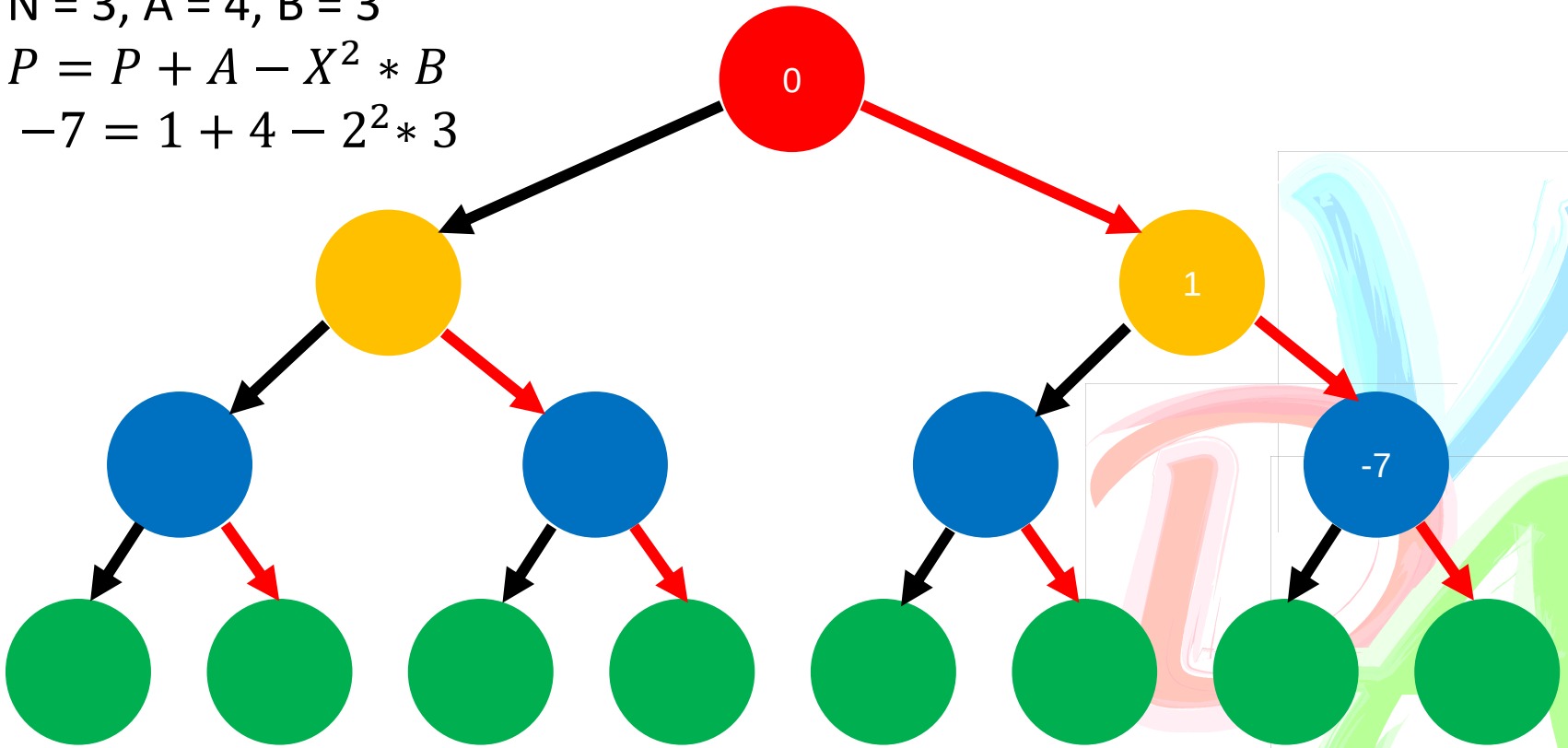
Rules

- Example (practice-> practice-> practice [first practice])
  - N = 3, A = 4, B = 3
  - $P = P + A - X^2 * B$
  - $1 = 0 + 4 - 1^2 * 3$



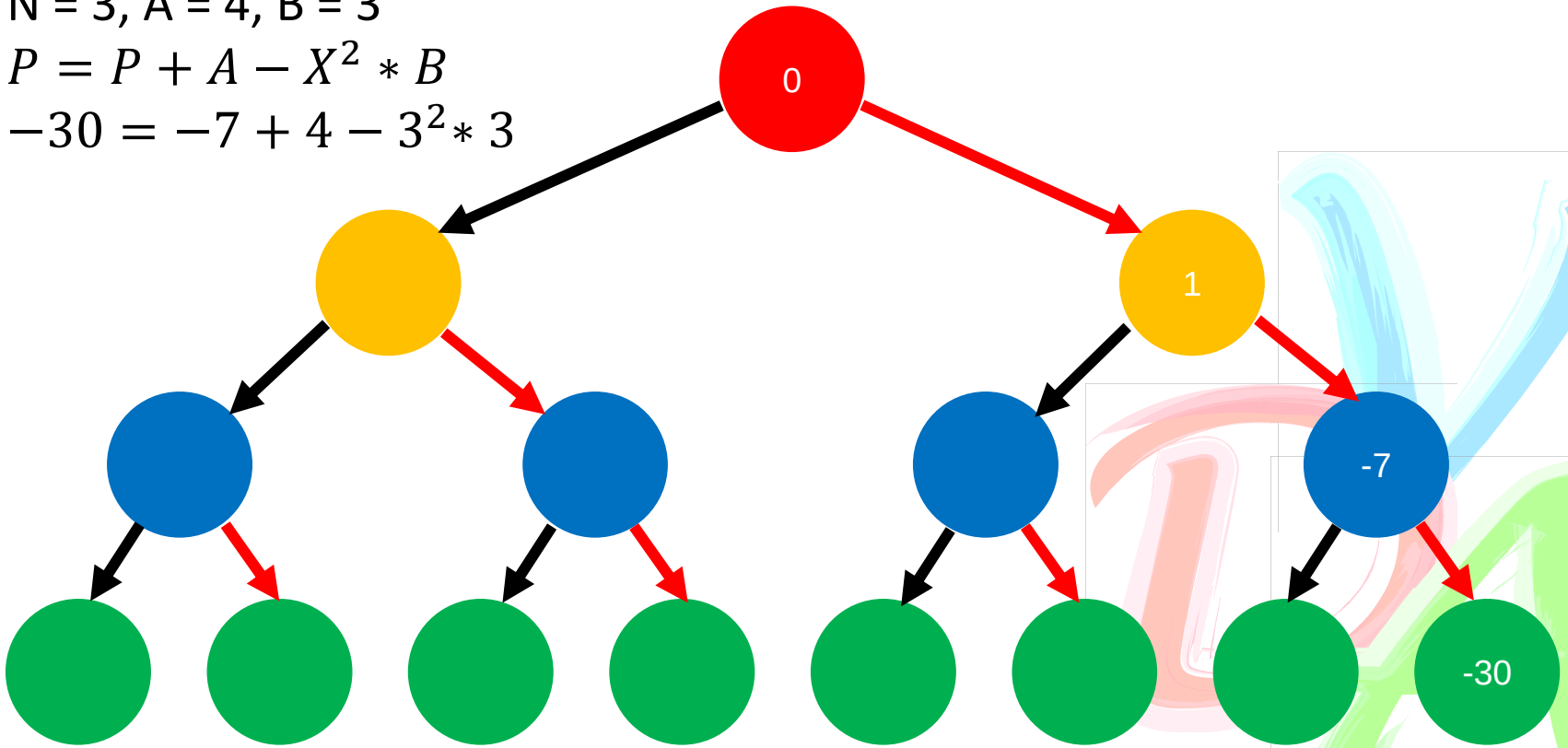
Rules

- Example (practice-> practice-> practice [second practice])
  - N = 3, A = 4, B = 3
  - $P = P + A - X^2 * B$
  - $-7 = 1 + 4 - 2^2 * 3$



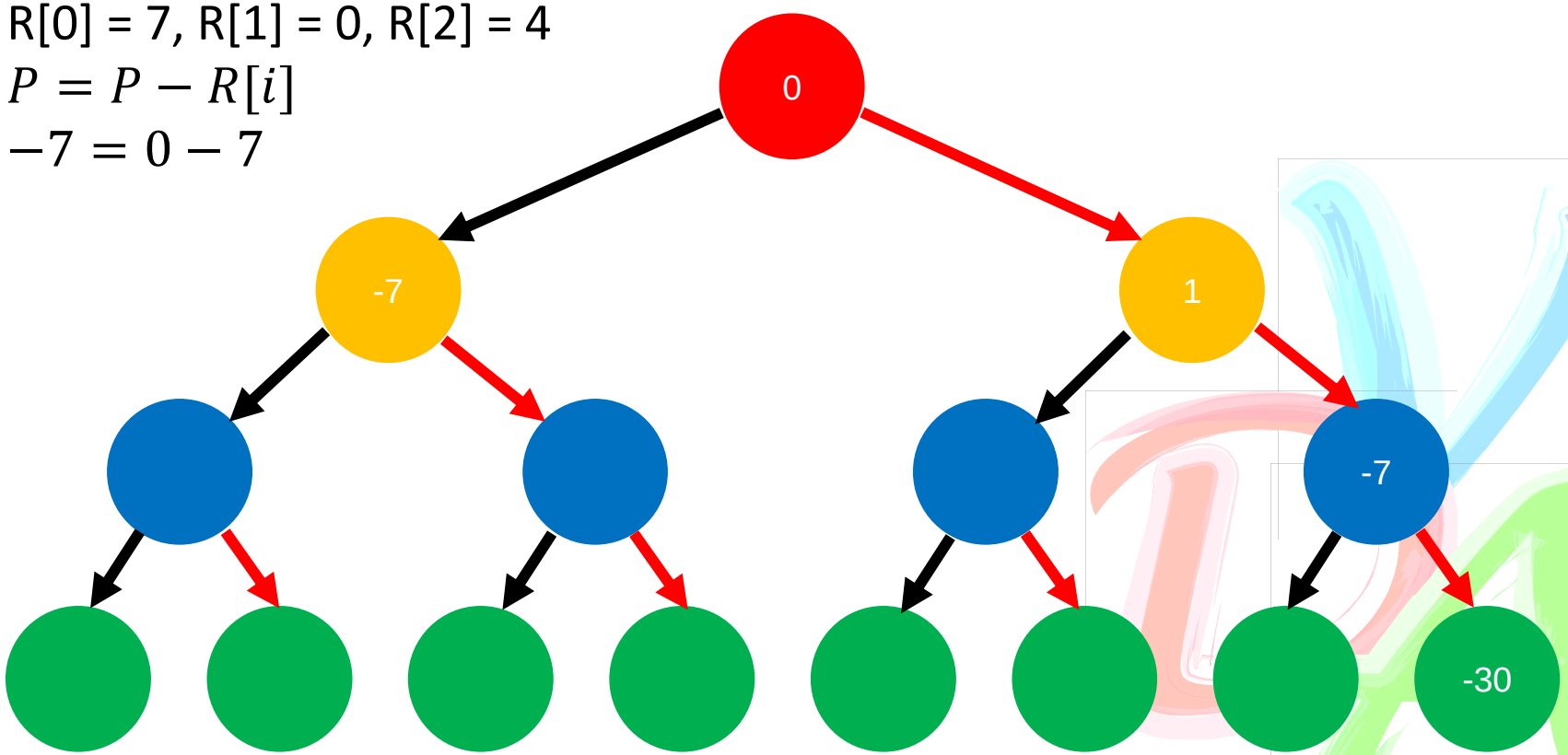
Rules

- Example (practice-> practice-> practice [third practice])
  - N = 3, A = 4, B = 3
  - $P = P + A - X^2 * B$
  - $-30 = -7 + 4 - 3^2 * 3$



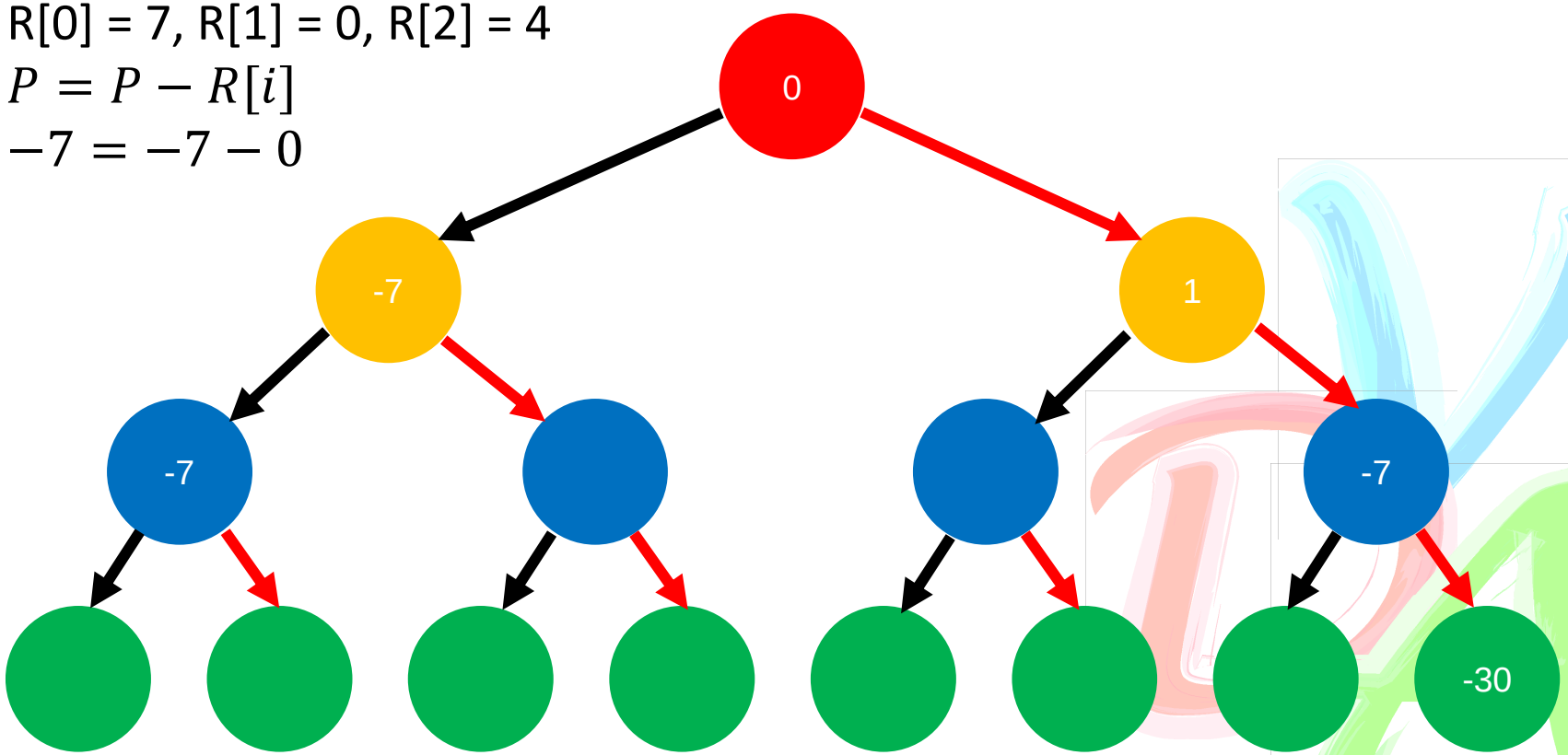
Rules

- Example (rest->rest->rest [first rest])
  - R[0] = 7, R[1] = 0, R[2] = 4
  - $P = P - R[i]$
  - $-7 = 0 - 7$



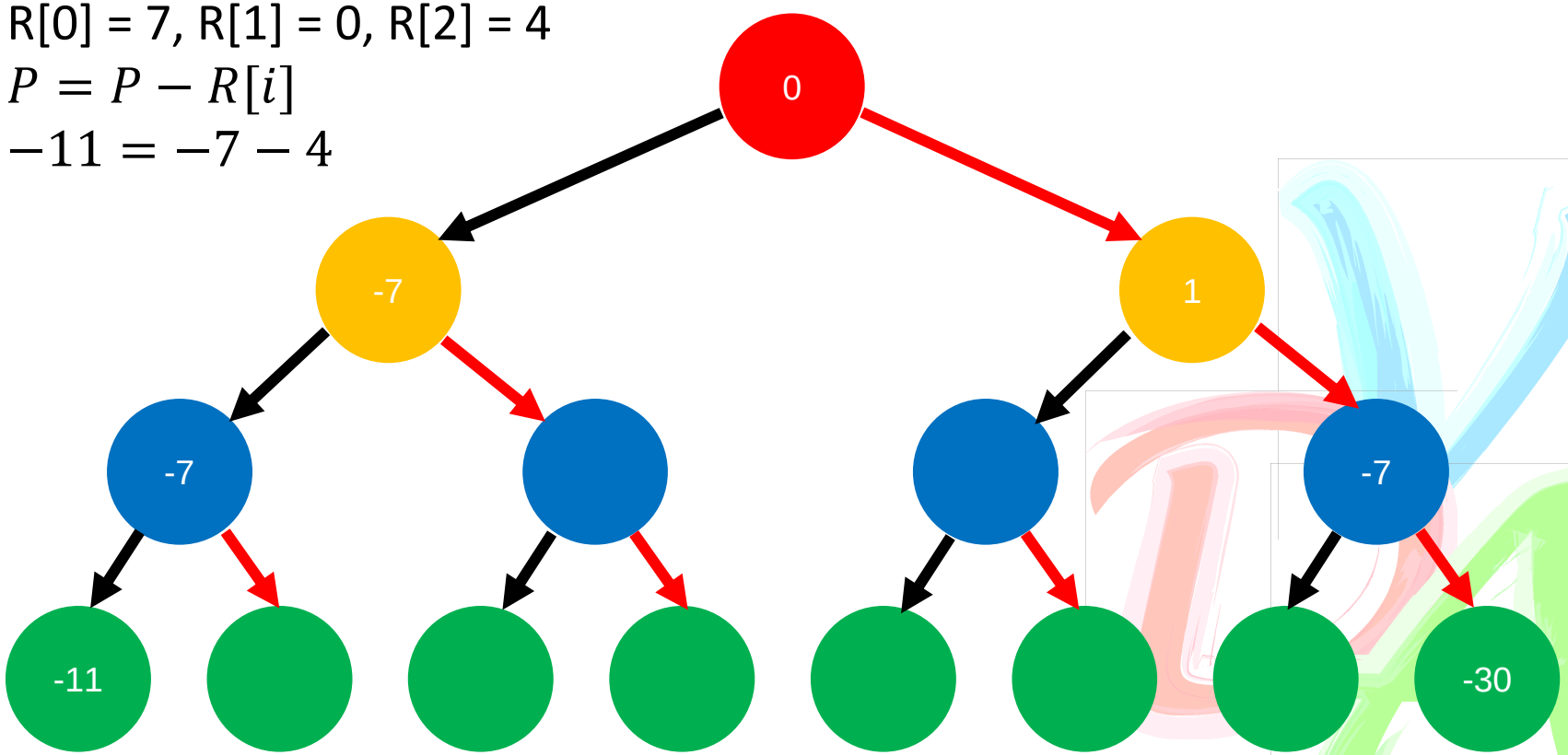
Rules

- Example (rest->rest->rest [second rest])
  - R[0] = 7, R[1] = 0, R[2] = 4
  - $P = P - R[i]$
  - $-7 = -7 - 0$



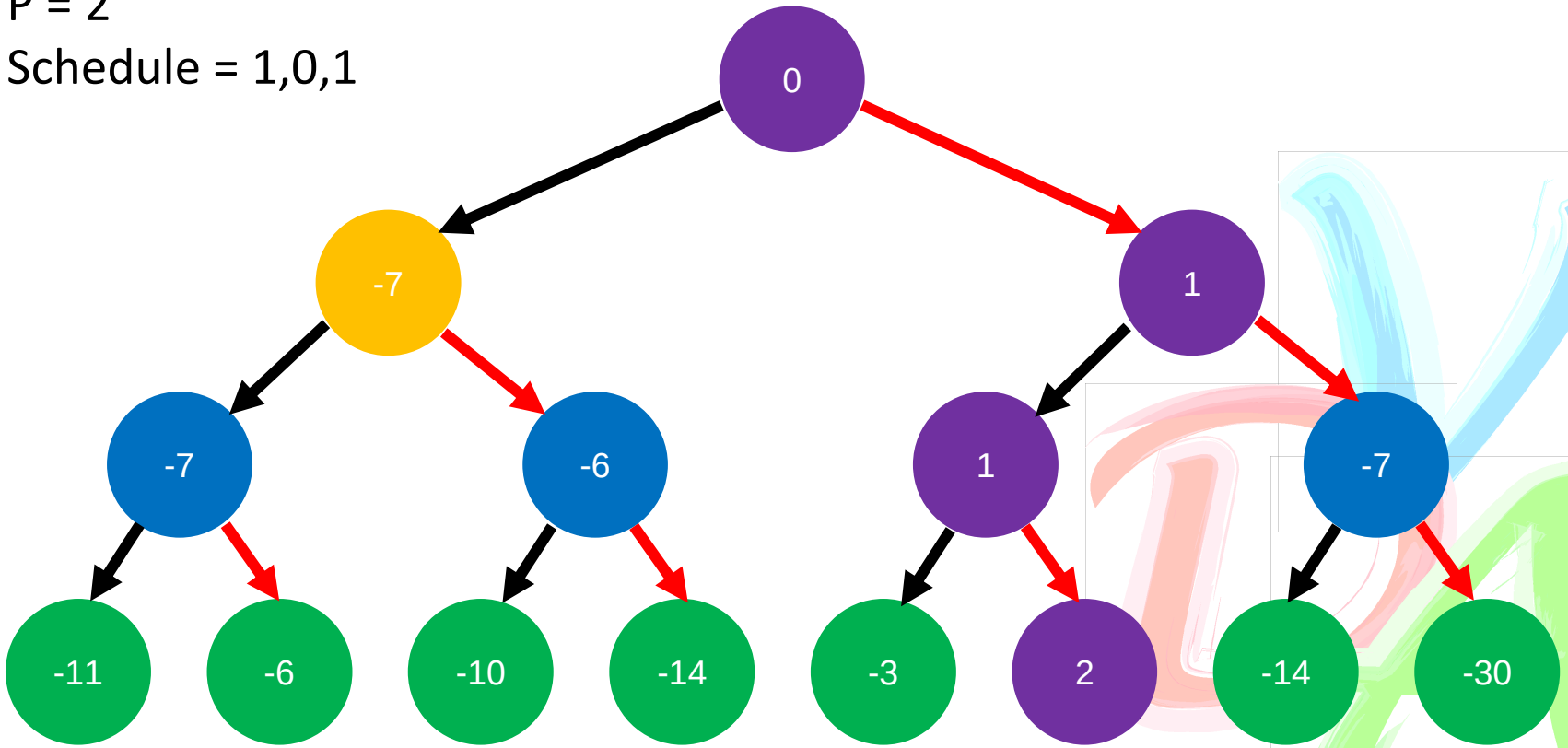
Rules

- Example (rest->rest->rest [third rest])
  - R[0] = 7, R[1] = 0, R[2] = 4
  - $P = P - R[i]$
  - $-11 = -7 - 4$



Rules

- Example (output)
  - P = 2
  - Schedule = 1,0,1



Input, Output format

- Input example

```
3 4 3   // N, A, B
7 0 4   // R[i]
```

  - First line represents the factors N, A, B.
  - Second line, it will give N numbers representing the R[0], R[i], … , R[N-1].

- Output example

```
2       // P
1 0 1   // 0 for rest, 1 for practice, from begin to end
```

  - First line output maximum performance
  - Second line output the rest and practice schedule of each day.

Specification

- Notes
  - Performance could be negative in the end.
  - All the factors in input file are integer.
  - The performance originally is 0.
  - If there are two or more ways to reach the maximum performance, select one of them will be correct.
  - The output maximum performance P could exceed $2^{31}-1$.

Grading

Time limit: 30 seconds for each case

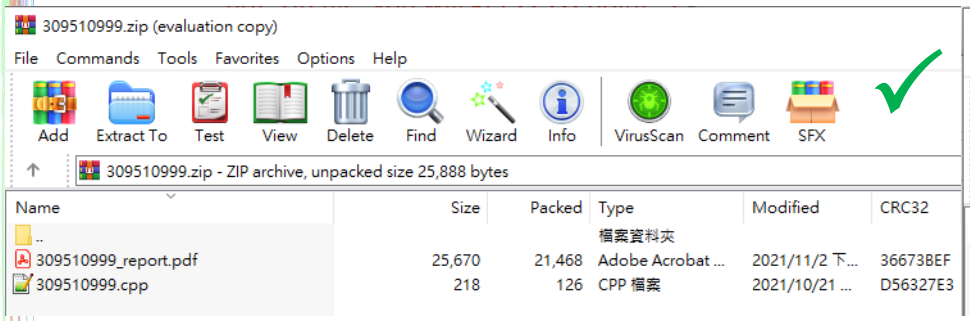
- Small Case (x5) $[10^0 \le N \le 10^4][0 \le A, B, R[i] \le 10^4]$ 50%
- Big Case (x3) $[10^4 \le N \le 10^6][0 \le A, B, R[i] \le 10^6]$ 30%
  - Correct answer 15%
  - Timing performance(if the answer is correct) 15%
- Report 20%
  - No more than 2 page
    i. Time complexity analysis
    ii. The flow chart of your program

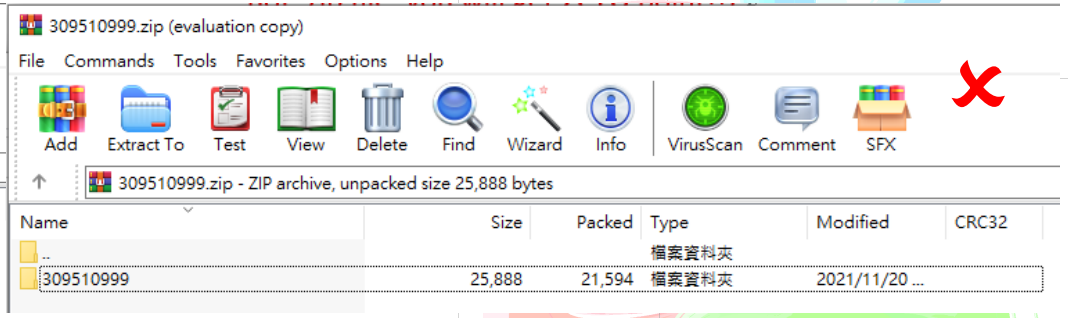
Submission

- <student_id>.zip *(example: 109511999.zip)*
  - Including source code and report
  - Source code: <student_id>.cpp *(example: 109511999.cpp)*
  - Report: **<student_id>_report.pdf** *(example: 109511999_report.pdf)*

Naming error: -5% per file

- zip format





Notice

- Please make sure your code is available on our linux server.
- Please use argc and argv to read input and output files.
- Do not print anything on the terminal!
- Please check the output format!
- Compile procedure: ***g++ -std=C++11 <student_ID>.cpp -o Lab1***
- Execution procedure: ***./Lab1 [input] [output]***
  - Example: *./Lab1 case1.txt output.txt*
- You **MUST WRITE YOUR OWN CODE**. Plagiarism is not allowed!!!

Thank you for listening.