



Problem D:

# Restrictive Filesystem

Yusuke Izumi

# 提出概況

- 提出数：59 (35 teams)
- 正答数：27
- 最初の正答
  - HITORI# (59分33秒) @東京大

# 問題概要

- (いかににも性能が悪そうな) ファイルシステムの以下の命令の挙動を再現せよ.
  - "W" I S
    - 識別子が I でサイズが S のファイルを書き込む.
  - "D" I
    - 識別子が I のファイルを削除する.
  - "R" P
    - P 番目のセクタを占有しているファイルの識別子を出力する.
- 書き出しの規則
  - 前方から順番に空き領域を探し出して, 片っ端から埋めていく.

# 解法

- 問題文を理解する。
  - 例を挙げて丁寧に説明がなされているので、おそらく読めば理解できるでしょう。
- 書かれたとおりに実装する。
  - まあ、これが大変だ、というわけですが。

# これは無理です

- セクタを全部展開する

0	1	2	3	...	$10^9-1$
2	2	1	1	...	空

- セクタの数は  $10^9$  (1G) 個ある
  - 整数を 1 個格納するには 4 バイトいるから...
  - 要するにメモリが 4GB ぐらい必要.
  - メモリがあつたとしても,  $10^9$  個の整数を書いたり消したりでは時間がかかりますよ.

# 正しい実装法

- 「範囲」と「識別子」を組にしたものの列を保持する。
  - 命令は 10000 個しかない。
  - よって、範囲と識別子の組は最大 10000 個にしかない。
  - (注) ファイルの削除によって空き領域が形成されるので、ひとつのファイルが分断することはあるが、それはもともと(前の命令で)別のファイルが占有していた領域にすぎない。

# 例

- Sample Input の最初の 4 個の命令を実行した直後であれば, 以下の要領で情報を保持する.
  - [ (0-1, 2) , (2-3, 1) , (4-5, 2) ]
  - (0-1, 2) は, 0 番目のセクタから 1 番目のセクタまでが File #2 で埋められている, という意味.

# クレジット

- 原作
  - 野田
- 本文
  - 黄
- 解答
  - 高橋, 泉
- 入力
  - 高橋