Problem E : Optimal Rest

原案 : 野田
模範解答 : 野田・北村
解説 : 野田

問題(1)

- MML表記された休符の長さを最適化せよ
  - 長さが同じ場合はASCIIコード順の辞書順で最も前に来るものを出力せよ

- 例:
  - “R2R2” → “R1”
  - “R1R2R4R8R16R32R64” → “R1.....”
  - “R1R4R16” → “R16R1R4”

問題(2)

- 文字列の長さは最大10万文字
- 使用可能な文字は”.”・”R”・”0”～”9”のみ
- 1/64休符単位未満の長さの休符は扱わない

惜しいけどダメな解法

- DP(動的計画法)
  - 縦方向に最後に使った休符の長さ、横方向にtick(1/64休符単位でカウントした時間)、各セルに文字列を持たせたDP
  - 文字列の長さをNと置くと計算量が$O(N^2)$となってしまいTLE

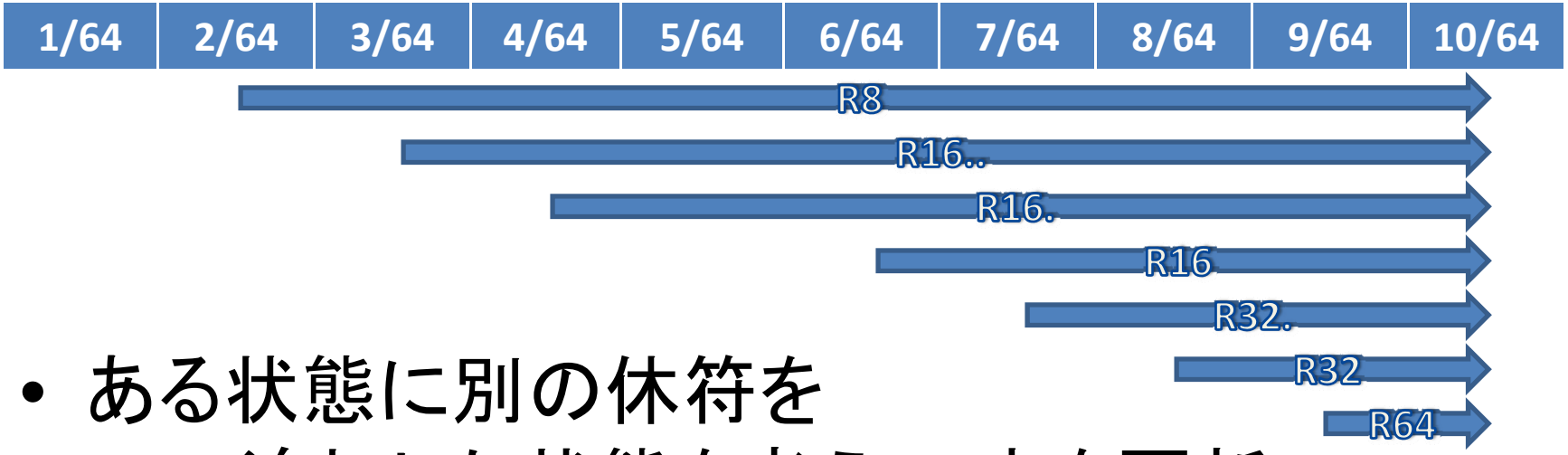
解法　その1 (1)

- 使用する各休符の個数を各セルに持たせた一次元DP
  - 休符の種類は28種類しかない
    - “R1”・”R1.”・”R1..”・・・”R32.”・”R64”
  - 各休符の使用個数が定まれば辞書順で最も前に来るMML表現も自ずと定まる
    - “R1.” × 4個 + ”R8” × 1個 + ”R32.” × 1個 = “R1.R1.R1.R1.R32.R8”

解法　その1 (2)

- 準備
  - 各休符を予め辞書順でソートしておく
    - ただし末尾に”R”をつけた状態でソート
      - 結合した時の辞書順を反映させるため

解法　その1 (3)



- ある状態に別の休符を一つ追加した状態を考えDP表を更新
  - MML表記の長さの総和が小さければ更新
  - MML表記の長さが同じで、辞書順で前に来る休符の個数が多ければ更新

解法　その1 (4)

- 最大文字列長・・・10万文字
- 最長休符表現・・・320万tick
  - “R1R1R1・・・”又は”R1.R1.・・・”が最も効率よく休符長を稼げる
  - 10万/(“R1”の文字列長)×(“R1”のtick)=320万tick

- 計算量・・・320万 × 28^2 = 2.5 × 10^9
  - 最初に計算すれば使い回しできる

解法2(1)

- 文字列末尾からのDP
  - 文字列の長さと先頭の休符の種類を保持
  - 更新時は文字列の長さと先頭の休符で判定

- 計算量O(N)
  - 文字列の長さ×休符の種類^2

解法　その3 (1)

- ダメ解答の~~チート~~改良版
- 途中までのテーブルを作成した後、規則性からprefixテーブルとsurfixテーブルを作成する

解法 その3 (2)

tick	MML
9/64	R64R8
9/64 + 96/64	R1.R64R8
9/64 + 96 / 64 × 2	R1.R1.R64R8
9/64 + 96 / 64 × 3	R1...R1..R1R64
9/64 + 96 / 64 × 4	R1...R1..R1.R1R64
9/64 + 96 / 64 × 5	R1...R1..R1.R1.R1R64
9/64 + 96 / 64 × 6	R1...R1..R1.R1.R1.R1R64

解法 その3 (3)

- 18拍(96×3tick)以降は6拍周期で似たMML表現が現れる
  - “R1.”の長さが6拍かつ辞書順で手前に来るため
  - 辞書順で“R1R1R1・・・”は” R1. R1. ・・・”より後

- 18拍+αまでは文字列を使ったDPでテーブル作成
- それ以降はテーブルからprefix/surfixテーブルを作成して出力

結果

- First Submit : LittleBug(89min) WA
- First Accepted : LittleBug(114min)
- Result : 5/43

ジャッジより

- 柔軟な思考が求められる問題です
- 応用力をつけましょう

御清聴有難うございました