# 古代の機械（Ancient Machine）

解説　戸高空

# 問題概要

- XYZ からなる長さNの文字列
- 文字列がなくなるまで、文字を1つずつ消していく
- X - - - Y - - Z のような並びでYを消す回数を最大化したい
- Anna … 文字列を知っている

  Brunoに0,1をLビット送る
- Bruno … 文字列を知らない

  文字を消す役割

# 制約

- N<=18（5点）
- N<=100000
  - L<=200000（累積 30点）
  - L<=160000（累積 40点）
  - L<=100000（累積 70点）
  - L<=70000　（満点）

# N<=18（5点）

- AnnaはBrunoに文字列の情報を全部送ることができる
- Brunoがよい消し方の最大値をO(2^N)くらいで達成できれば良い
- dp[S] … 残っている集合がSのときのよい消し方の最大値でbitDP
- dpテーブルをもとに最大値を達成する消し方を復元できる

# L<=200000（累積 30点）

- AnnaはBrunoに文字列の情報を全部送ることができる
- X→00,Y→01,Z→10に変換して1文字につき2ビット使って送る
- Brunoがよい消し方の最大値をO(N)くらいで達成できれば良い

# よい消し方の最大値

- 一番左のXより左の文字はよい消し方はできない
- 一番右のZより右の文字はよい消し方はできない
- 連続するYはそのうち1つしかよい消し方はできない
- 最大値は一番左のXと一番右のZの間にある連続するYの区間の個数以下になる
- 実際にこれは達成できる

# 最大値の達成の仕方

- 一番左のX（場所をaとする）より左の文字は消す
- 残った文字で一番左のZを見る（場所をbとする）
- aとbの間の文字はXかYのみである
- bの一つ左を順番に消していく
- 連続したYの中で一番左のYは必ずよい消し方になる
- 最後にbを消す
- 同じように残った文字で一番左のZを見ていく

# 例

YZXYYZZYXXYZZZ

--XYYZZYXXYZZZ

--XY-ZZYXXYZZZ

--X--ZZYXXYZZZ good

--X---ZYXXYZZZ

--X----YXXYZZZ

--X----YXX-ZZZ good

# 例

--X----YXX-ZZZ

--X----YX--ZZZ

--X----Y---ZZZ

--X--------ZZZ good

--X---------ZZ

--X----------Z

--X-----------

# L<=160000（累積 40点）

- AnnaはBrunoに文字列の情報を全部送る
- 文字は3種類しかないので1文字に2ビット使うのは無駄
- 文字列の種類は$3^N$通りしかない
- 理論的には$\log_2 3^N \fallingdotseq 1.58N$ビットで十分
- まとめて送るほど理論値近づく
- $3^5 \leq 2^8$より5文字ずつを8ビットで送ると
  160000ビットを達成できる

# L<=100000（累積 70点）

- Brunoが最大値を達成するために必要な情報のみ送る

# L<=100000（累積 70点）

- 最大値の達成の仕方
- 一番左のX以降で、左のZから順にZの左隣を消していく
- 必要な情報は一番左のXとそれ以降のZの位置
- 各文字1ビット使って送ることができる
- 一番左のXまたはそれ以降のZなら1、それ以外なら0を送る
- 例：YZXYYZZYXXYZZZ

  00100110000111

# L<=100000（累積 70点）

- 最大値の達成の仕方
- 一番左のX以降で、左のZから順にZの左隣を消していく
- 必要な情報は一番左のXとそれ以降のZの位置
- 各文字1ビット使って送ることができる
- 一番左のXまたはそれ以降のZなら1、それ以外なら0を送る
- 例：YZXYYZZYXXYZZZ

  00100110000111

# L<=70000（満点）

- 先ほどの解法ではNビットのあらゆるビット列が登場する
- アルゴリズムを変えないとこれ以上減らせない

# 最大値の達成の仕方（改）

- 一番左のX（場所をaとする）より左の文字は消す
- 残った文字で一番左のZを見る（場所をbとする）
- ただしそのZが連続している場合はその中で一番右のZを見る
- bの一つ左を順番に消していく
- 連続したYの中で一番左のYは必ずよい消し方になる
- 最後にbを消す
- 同じように残った文字で一番左のZを見ていく

# L<=70000（満点）

- このようにアルゴリズムを変えても最大値を達成できる
- またこの改善によって送るべきビット列に次の性質が生まれた
- 「1は連続して現れない」
- 登場するビット列が制限された

# L<=70000（満点）

- 「1が連続して現れない」Nビット列は$2^N$種類より少ない
- 「1が連続して現れない」ビット列に辞書順で番号をつけてその番号を2進数で送ればよい
- Bruno側では辞書順で(送られてきた番号)番目の「 1が連続して現れない」ビット列を求めて復元する
- 各$i$について「1が連続して現れない$i$ビット列の数」を計算しておくとできる
- 実装の都合上、番号がオーバーフローしないように、何ビットかずつに分けて送る

# L<=70000（満点）

- 「1が連続して現れない」Nビット列は何通りあるだろうか
- $dp[i][0]$…$i$文字目が0のとき$i$ビット列は何通りあるか

  $dp[i][1]$…$i$文字目が1のとき$i$ビット列は何通りあるか

  $dp[i][0] = dp[i-1][0] + dp[i-1][1]$

  $dp[i][1] = dp[i-1][0]$だから

  $dp[i][0] = dp[i-1][0] + dp[i-2][0]$

これはフィボナッチ数列

# L<=70000（満点）

- 「1が連続して現れない」Nビット列は$fib(N+2)$種類ある
- ただし$fib(i)$はフィボナッチ数列の$i$項目を表す
- 理論的には$\log_2 fib(N+2) \fallingdotseq 0.69N$ビットで送ることができる
- 例えば$fib(63+2) \leq 2^{44}$なので63ビットずつを44ビットで送ると69872ビットが達成できる

# 最大値の達成の仕方(別解)

- 一番左のX（場所をaとする）より左の文字は消す
- aの右隣りの文字を見ていく
- Xのとき…消してよい
- Yのとき…その次がXなら、このYXを残して次の文字をみる
  そうでなければ消してよい
- Zのとき…消してよい（最後のZの場合は残しておいた文字を後ろから消していく）

# 最大値の達成の仕方(別解)

- これで最大値が達成できる
- 必要な情報は最初のX、それ以降のYXの並び、最後のZ
- これを各文字ごとに0or1で送ると100000ビット
- 例：YZXYYZZYXXYXZZZX

  0010000100100010
- またYXで送る1は連続しないのでさきほどと同様に満点もとれる

# 得点分布