「場合の数」とデータ圧縮

東京大学
大学院情報理工学系研究科数理情報学専攻
工学部計数工学科
定兼 邦彦

2014年8月6日

http://researchmap.jp/sada/

自己紹介

- 名前: 定兼　邦彦　（さだかね　くにひこ）
- 所属: 東京大学大学院情報理工学系研究科数理情報学専攻
- 経歴: 1991年理1入学, 2000年理学系研究科情報科学専攻修了
- 2000年　東北大学大学院情報科学研究科助手
- 2003年　九州大学大学院システム情報科学研究院助教授
- 2009年　国立情報学研究所准教授
- 2014年　東京大学大学院情報理工学系研究科教授
- 研究分野: データ圧縮, データ構造, 情報検索

- 最近の研究:
- ビッグデータ処理
- 簡潔データ構造
- 並列アルゴリズム

「場合の数」とデータ圧縮

- 場合の数とは
  - ある事柄の起こり方の総数 （大辞林より）
  - 例：1から5の数字から2つの数字を選ぶやり方は何通りあるか．答: ${}_5C_2 = 10$ 通り．
- データ圧縮とは
  - 音楽，動画など，データ量の多いものを小さくする
- 両者にどのような関係が？

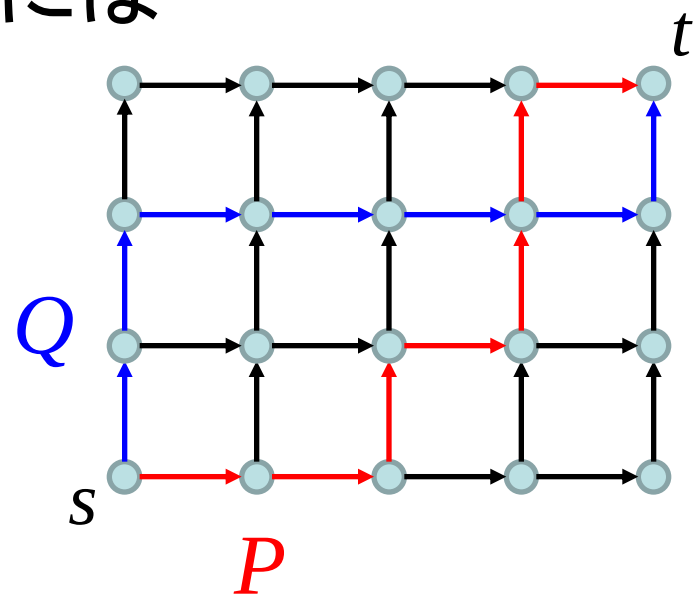
いろいろな圧縮

- 空き缶，ペットボトルをつぶす
  - 元には戻らない
- 布団圧縮袋
  - 圧縮できるが元に戻すには時間がかかる
- 音楽，画像，動画の圧縮
  - 人が気づかないところは消してしまう
- 通常のファイル圧縮 (zipなど)
  - 布団圧縮袋と同じ
- データを圧縮すると，そのままでは使えない

ビッグデータ

- お店での購買履歴，人の移動履歴，DNA配列情報など，様々な大量データが存在
- ビッグデータを処理するには大容量メモリが必要
  - スマホ:1 GB (ギガバイト)
  - ノートPC，デスクトップPC: 4 ~ 32GB
  - 人のDNA配列を読み取る処理で必要なメモリ: 300GB
- データを圧縮したまま処理できるとうれしい

- s から t へ矢印の通りに移動するルートを表現するには

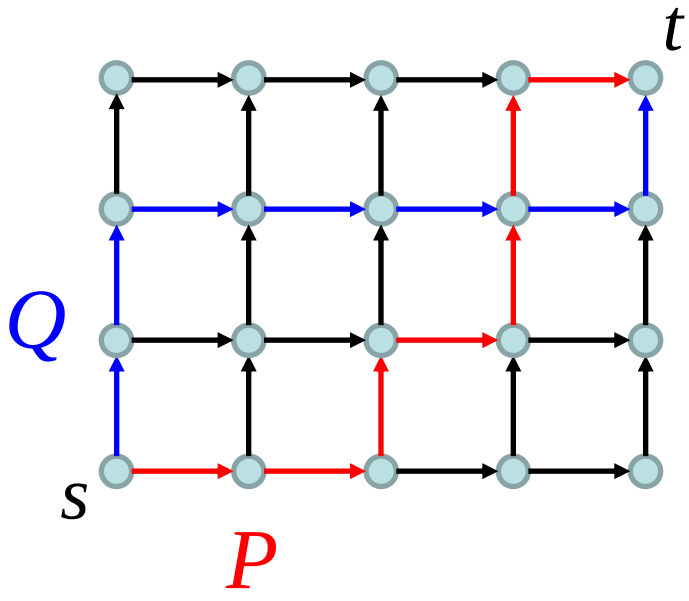


P: → → ↑ → ↑↑ → 0010110

Q: ↑↑ → → → → ↑ 1100001

7ビットで表現できる

- s から t へのルートは集合を表している



P: → → ↑ → ↑↑ → 0010110 {3, 5, 6}

Q: ↑↑ → → → → ↑ 1100001 {1, 2, 7}

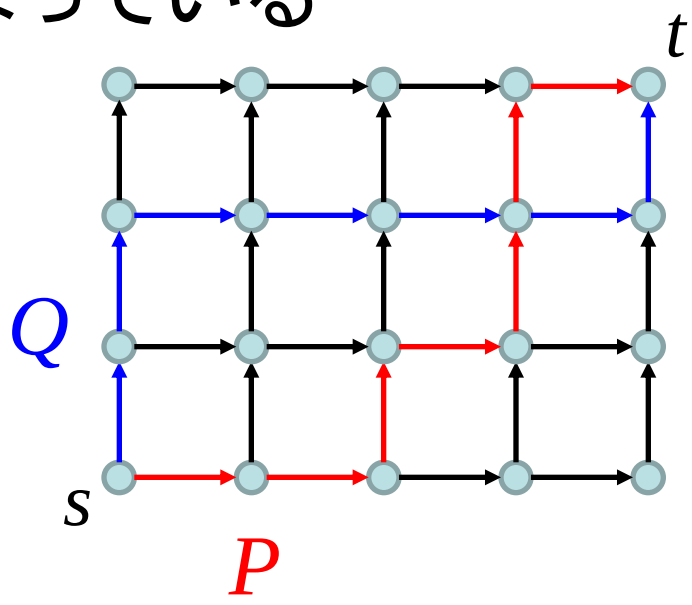
- 1 から 7 の数字から 3 つ選んだ集合を表す
- 集合が圧縮できている
- もっと圧縮できる?

- s から t へ矢印の通りに移動する行き方は何通り?
- ルートは4個の→と3個の↑で表現できる
- 逆に，4個の→と3個の↑をどういう順に並べても，それは s から t へのルートになっている
- 行き方は

$$ {}_7C_3 = {}_7C_4 = 35 \text{ 通り} $$

$$ \binom{7}{3} = \binom{7}{4} = 35 $$

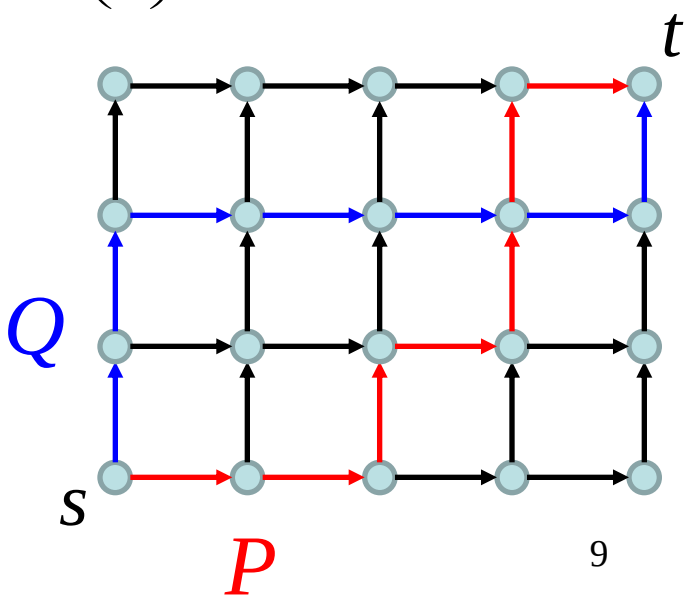
- $2^6 = 64 > 35$ なので，6 ビットで表現できる



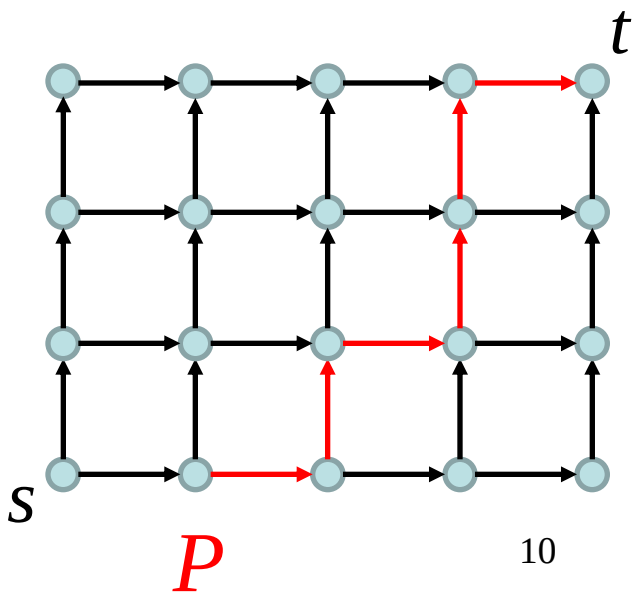
P: → → ↑ → ↑ ↑ →

Q: ↑ ↑ → → → → ↑

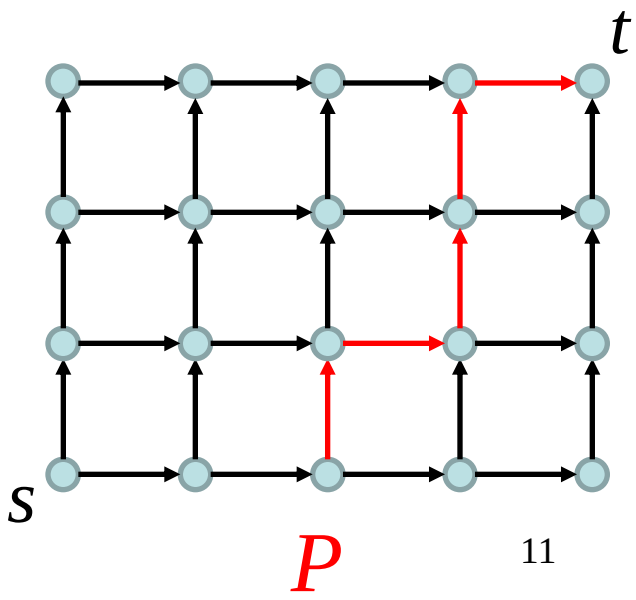
- s から t へのルートを 6 ビットで表現するにはどう符号化すればいいか
- 各ルートを 0 から 34 の整数で表現する
- s から最初に右に行くルートは $\binom{6}{3} = 20$ 通り
  - 0 から 19 の整数で表現する
- s から最初に上に行くルートは $\binom{6}{2} = 15$ 通り
  - 20 から 34 の整数で表現する



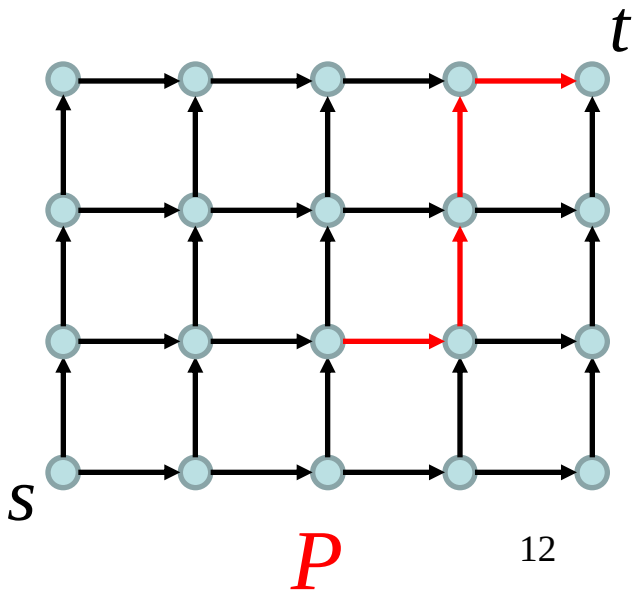
- s から最初に右に行くルート20個を考える
- 次に右に行くルートは $\binom{5}{3}=10$ 通り
  - 0 から 9 の整数で表現する
- 次に上に行くルートは $\binom{5}{2}=10$ 通り
  - 10 から 19 の整数で表現する



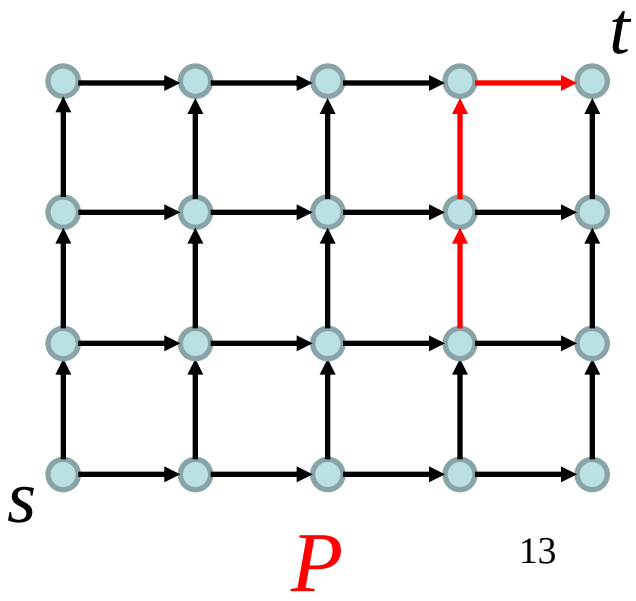
- s から右，右に行くルート10個を考える
- 次に右に行くルートは $\binom{4}{3} = 4$ 通り
  - 0 から 3 の整数で表現する
- 次に上に行くルートは $\binom{4}{2} = 6$ 通り
  - 4 から 9 の整数で表現する



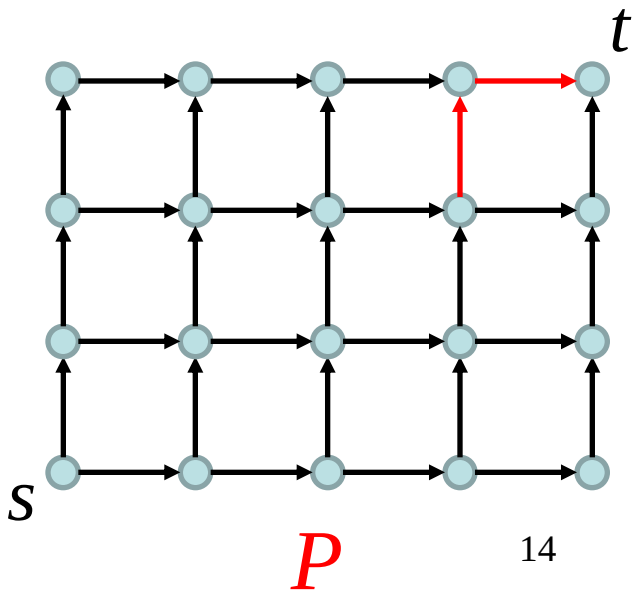
- s から右右上に行くルート6個を考える
- 次に右に行くルートは $\binom{3}{2}=3$ 通り
  - 4 から 6 の整数で表現する
- 次に上に行くルートは $\binom{3}{1}=3$ 通り
  - 7 から 9 の整数で表現する



- s から右右上右に行くルート3個を考える
- 次に右に行くルートは $\binom{2}{2}=1$ 通り
  - 4 から 4 の整数で表現する
- 次に上に行くルートは $\binom{2}{1}=2$ 通り
  - 5 から 6 の整数で表現する



- s から右右上右上に行くルート2個を考える
- 次に右に行くルートは $\binom{1}{1}=1$ 通り
  - 5 から 5 の整数で表現する
- 次に上に行くルートは $\binom{1}{0}=1$ 通り
  - 6 から 6 の整数で表現する

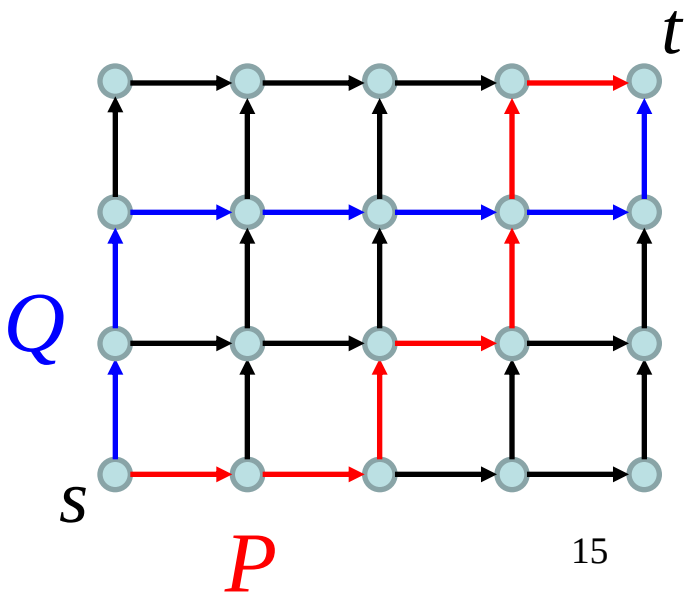


- 結局，経路 P は整数 6 で表される

$$\binom{4}{3}+\binom{2}{2}+\binom{1}{1}=4+1+1=6$$

- 同様に，経路 Q は整数 30 で表される

$$\binom{6}{3}+\binom{5}{2}+\binom{0}{1}=20+10+0=30$$



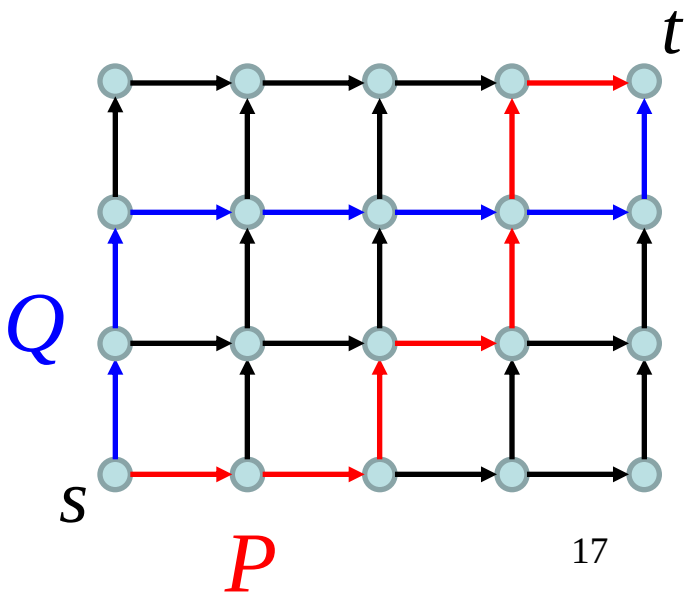
- 一般に，上に r 回，右に $n–r$ 回移動する場合，ルートの数は $\binom{n}{r}$
- 最初に右に行くルートの数は $\binom{n-1}{r}$
- 最初に上に行くルートの数は $\binom{n-1}{r-1}$

$$\binom{n}{r}=\binom{n-1}{r}+\binom{n-1}{r-1}$$

- n 個の矢印のうち，↑の位置を $n_r, n_{r-1}, \ldots, n_1$ とするとルートは整数 $\binom{n_r}{r}+\binom{n_{r-1}}{r-1}+\Lambda+\binom{n_1}{1}$ で表される

ルートの復元

- ルートを表す整数 30 から，ルートを復元する
- s から最初に右に行くルートは $\binom{6}{3} = 20$ 通り
- 30>20 なので，最初は上に行っている
- 最初に上に行った点からルート 30–20 = 10 を復元

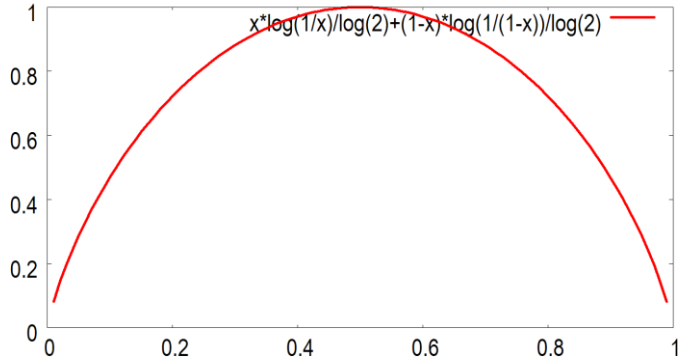


圧縮率

- 上に r 回，右に $n-r$ 回移動する場合
  - n 個の矢印→↑…で表現: n ビット
  - 整数に変換して表現: $\left\lceil \log_2 \binom{n}{r} \right\rceil$ ビット
- $\left\lceil \log_2 \binom{n}{r} \right\rceil \le \left\lceil \log_2 2^n \right\rceil = n$ なので，変換する方が小さい

$$\left\lceil \log_2 \binom{n}{r} \right\rceil \approx r \log_2 \frac{n}{r} + (n-r) \log_2 \frac{n}{n-r}$$

$$= nH\left(\frac{r}{n}\right)$$ エントロピー

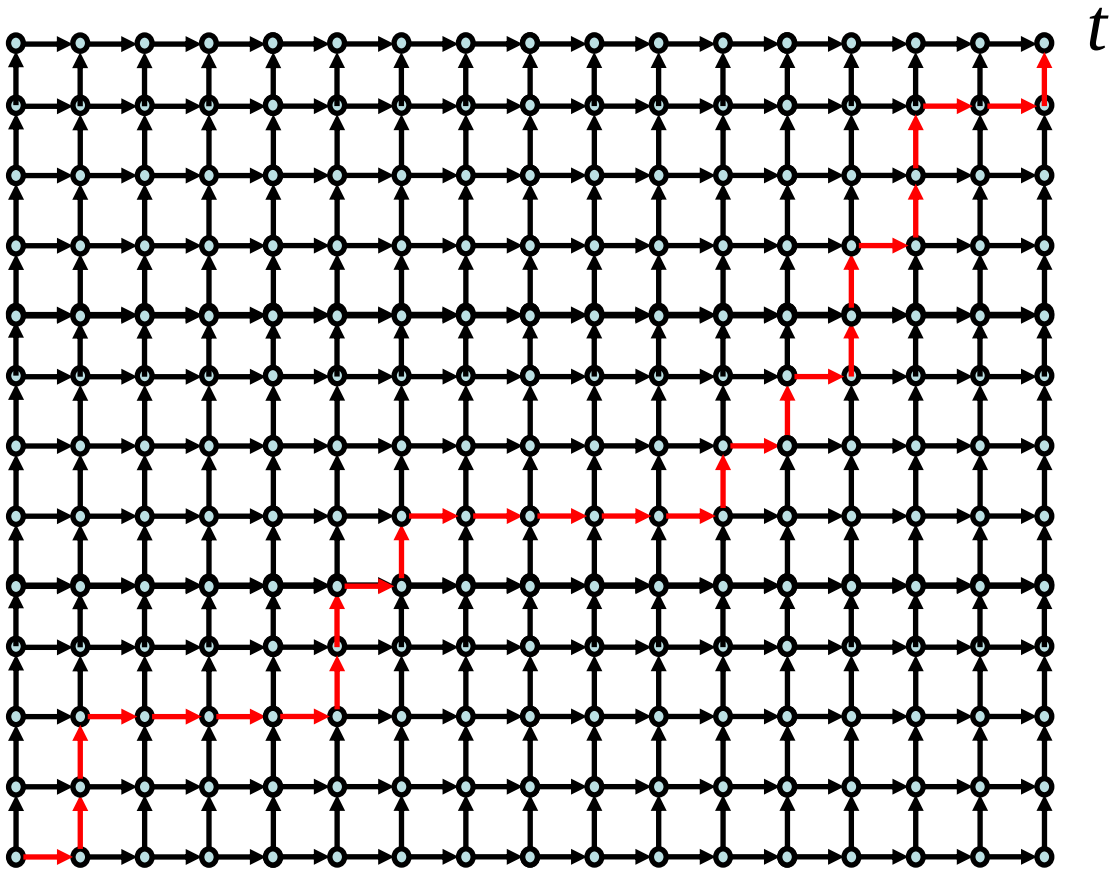


なので，r が小さければ圧縮率が高くなる
(もしくは $n-r$ が小さければ)

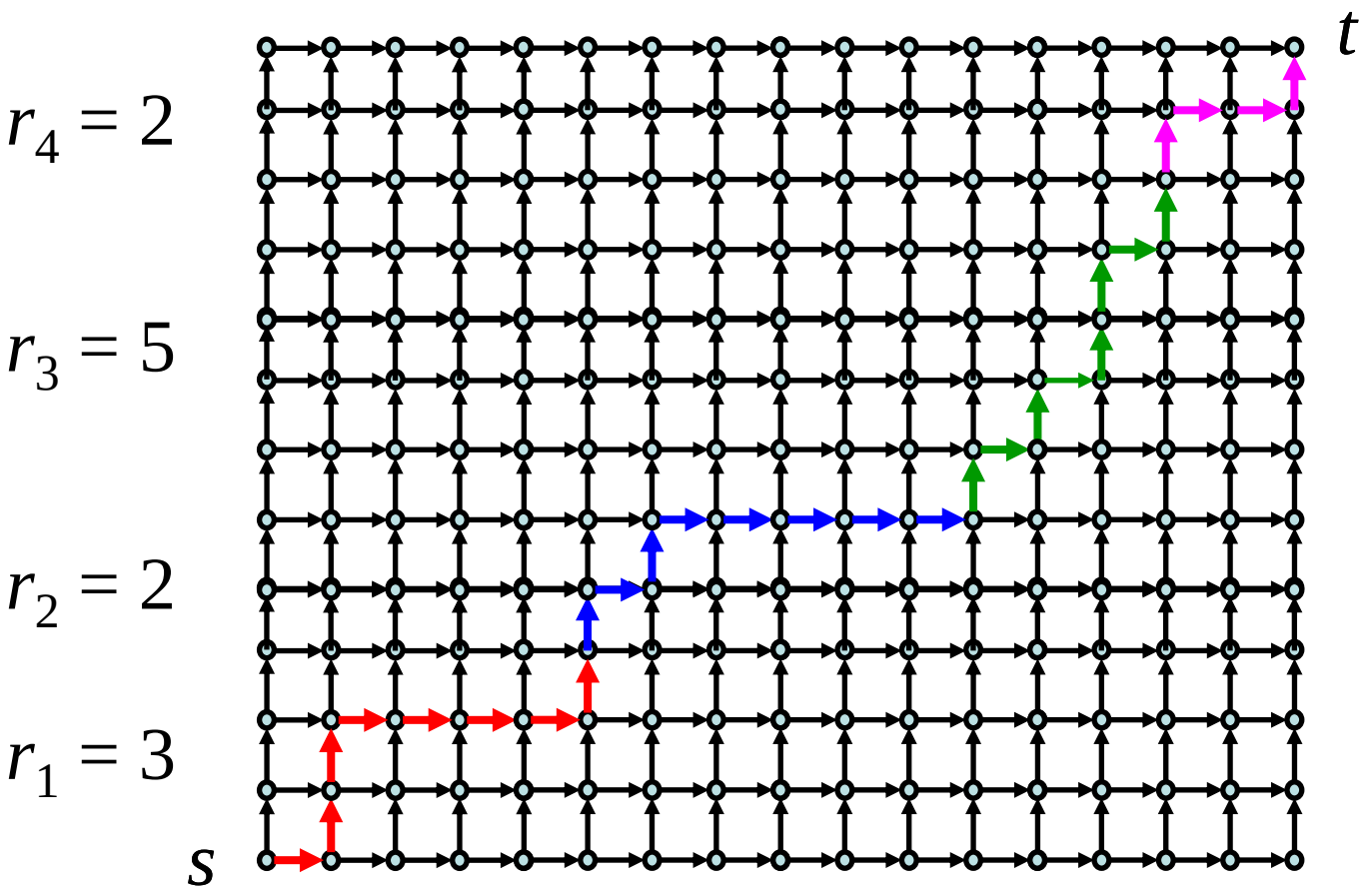
- 厳密には，ルートを表す $\left\lceil \log_2 \binom{n}{r} \right\rceil$ ビットの値の他に n と r の値も保存する必要がある
- 合計で $\left\lceil \log_2 \binom{n}{r} \right\rceil + 2\lceil \log_2 n \rceil + 2\lceil \log_2 n \rceil + 2$ ビット

n が大きい場合

- $n > 10$億
- $\binom{n}{r}$ を計算するには，大きい数の掛け算，割り算が必要になり，遅いしメモリも多く必要となる

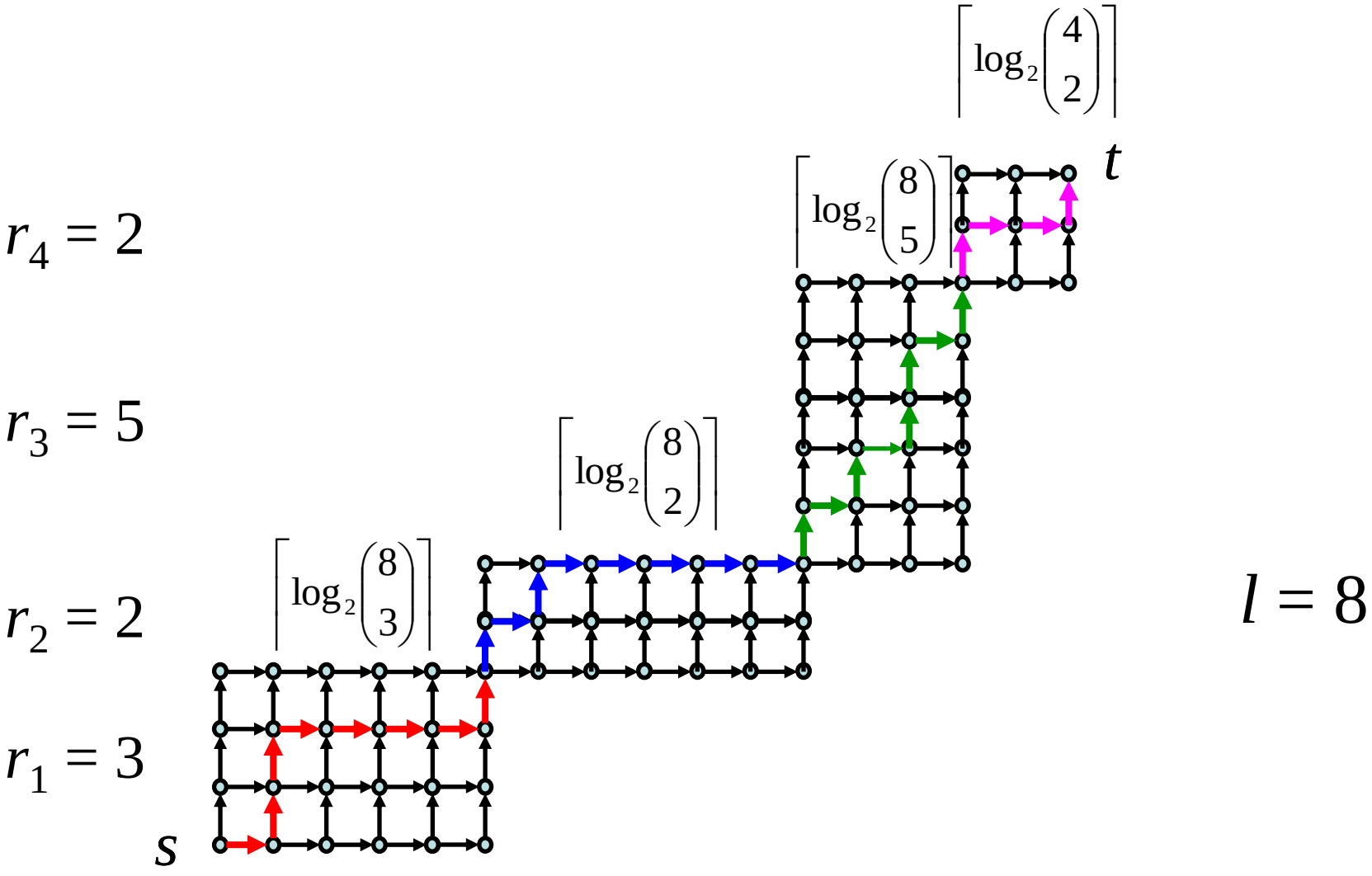


- ルートを一定の長さ l ごとに区切る
- 各部分ルートに対し，↑の数を記録する



$l = 8$

- 各部分ルートは $\left\lceil \log_2 \binom{l}{r_i} \right\rceil$ ビットで表現できる



- $\left\lceil \log_2 \binom{l}{r_1} \right\rceil + \left\lceil \log_2 \binom{l}{r_2} \right\rceil + \left\lceil \log_2 \binom{l}{r_3} \right\rceil + \Lambda$ はどれくらいの大きさ?

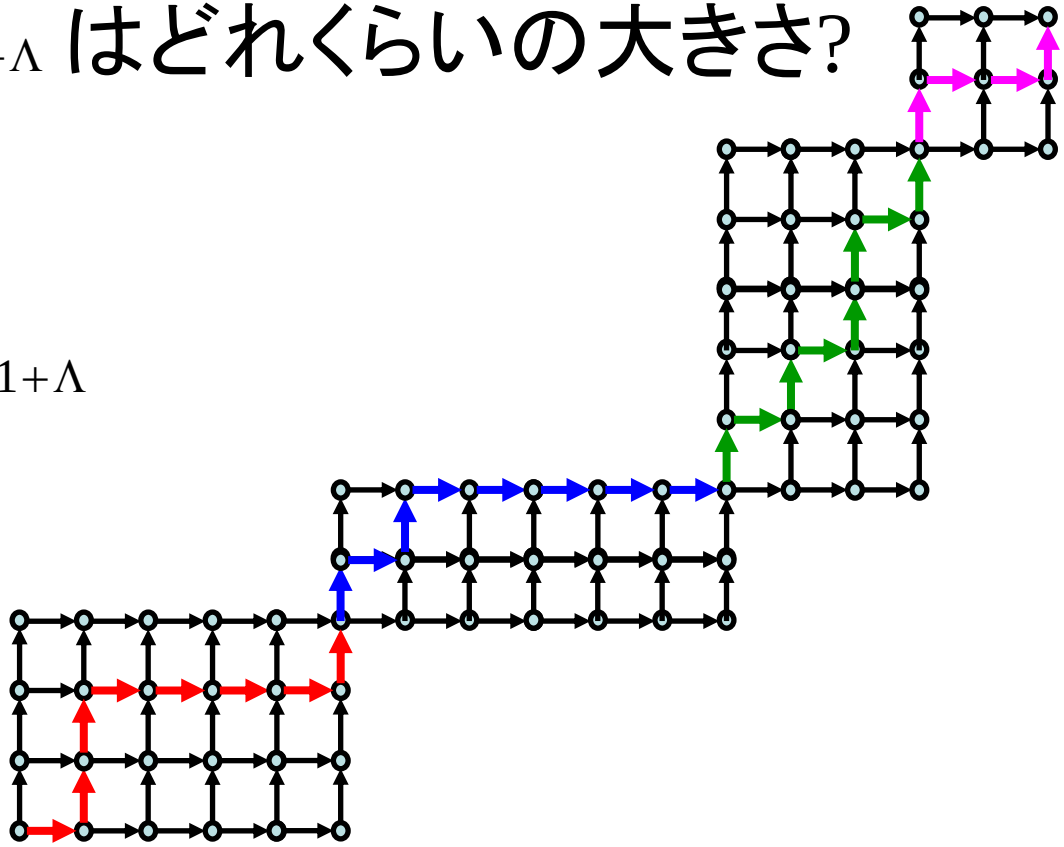
$$\left\lceil \log_2 \binom{l}{r_1} \right\rceil + \left\lceil \log_2 \binom{l}{r_2} \right\rceil + \left\lceil \log_2 \binom{l}{r_3} \right\rceil + \Lambda$$

$$< \log_2 \binom{l}{r_1} + 1 + \log_2 \binom{l}{r_2} + 1 + \log_2 \binom{l}{r_3} + 1 + \Lambda$$

$$= \log \left(\binom{l}{r_1} \cdot \binom{l}{r_2} \cdot \binom{l}{r_3} \cdot \Lambda \right) + \left\lceil \frac{n}{l} \right\rceil$$

$$\leq \log \binom{l+l+l+\Lambda}{r_1+r_2+r_3+\Lambda} + \left\lceil \frac{n}{l} \right\rceil$$

$$= \log \binom{n}{r} + \left\lceil \frac{n}{l} \right\rceil$$



集合 {2, 3, 8, 9, 11, 17, 19, 21, 22, 24, 25, 28} を表す

- 分割するとサイズは小さくなる
- 一部分だけを高速に復元できる

  ⇒ 全体は圧縮したまま使える

かな漢字変換辞書

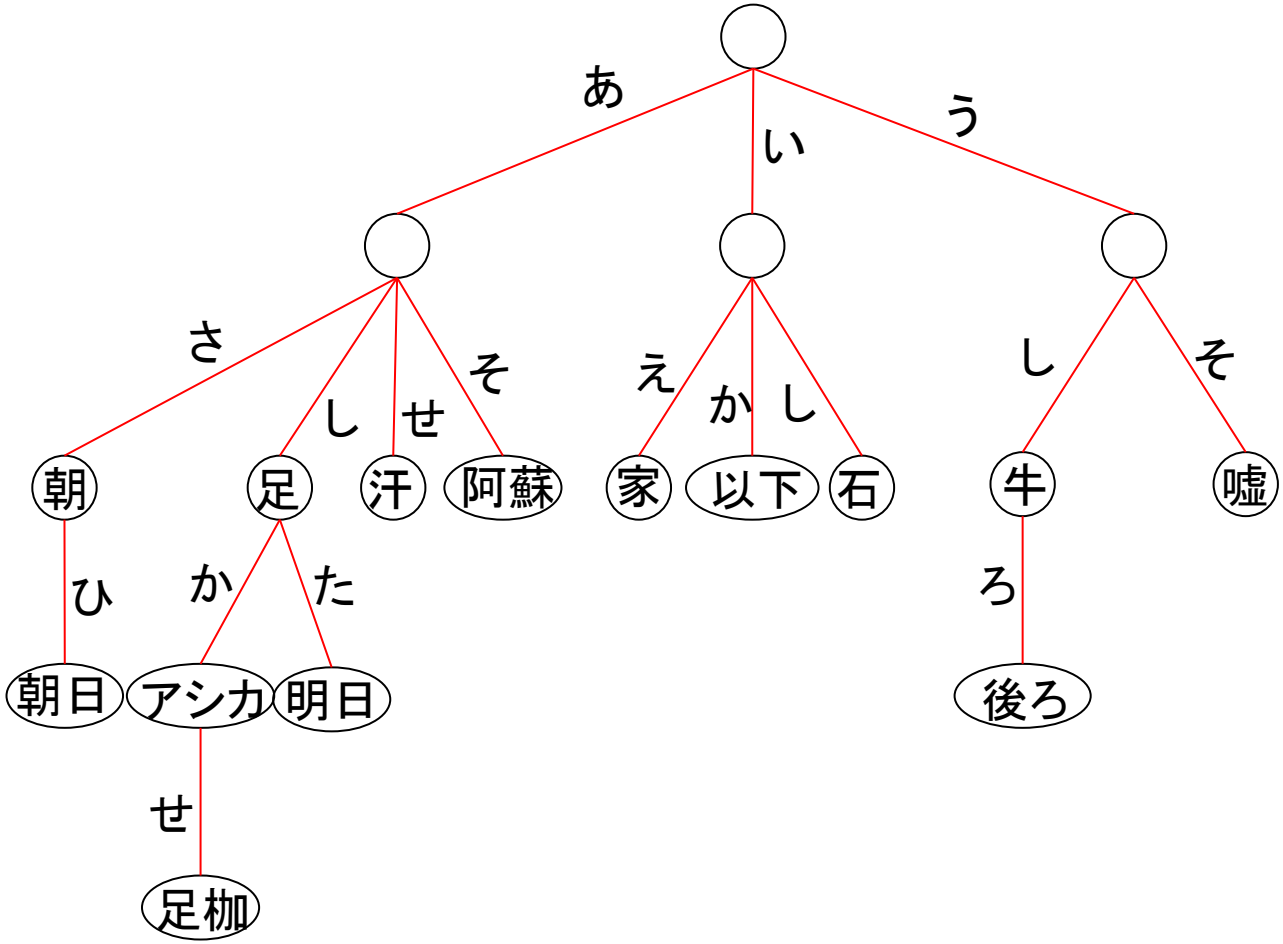
- 「読み」の辞書順に従って単語を木に格納する
- 同じ文字が何度も出てくるので圧縮したい

辞書

あさ → 朝
あさひ → 朝日
あし → 足
あしか → アシカ
あしかせ → 足枷
あした→ 明日
あせ → 汗
あそ → 阿蘇
いえ → 家
いか → 以下
いし → 石
うし → 牛
うしろ → 後ろ
うそ → 嘘

- 接頭辞 (prefix) が同じ部分は1つにまとめる

⇒トライ (trie) と呼ばれる木構造

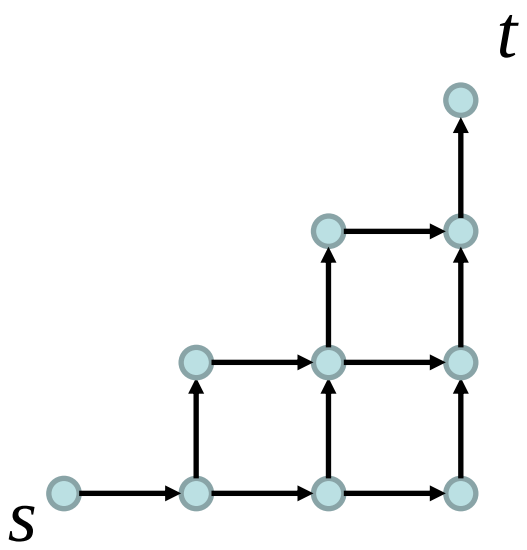


辞書

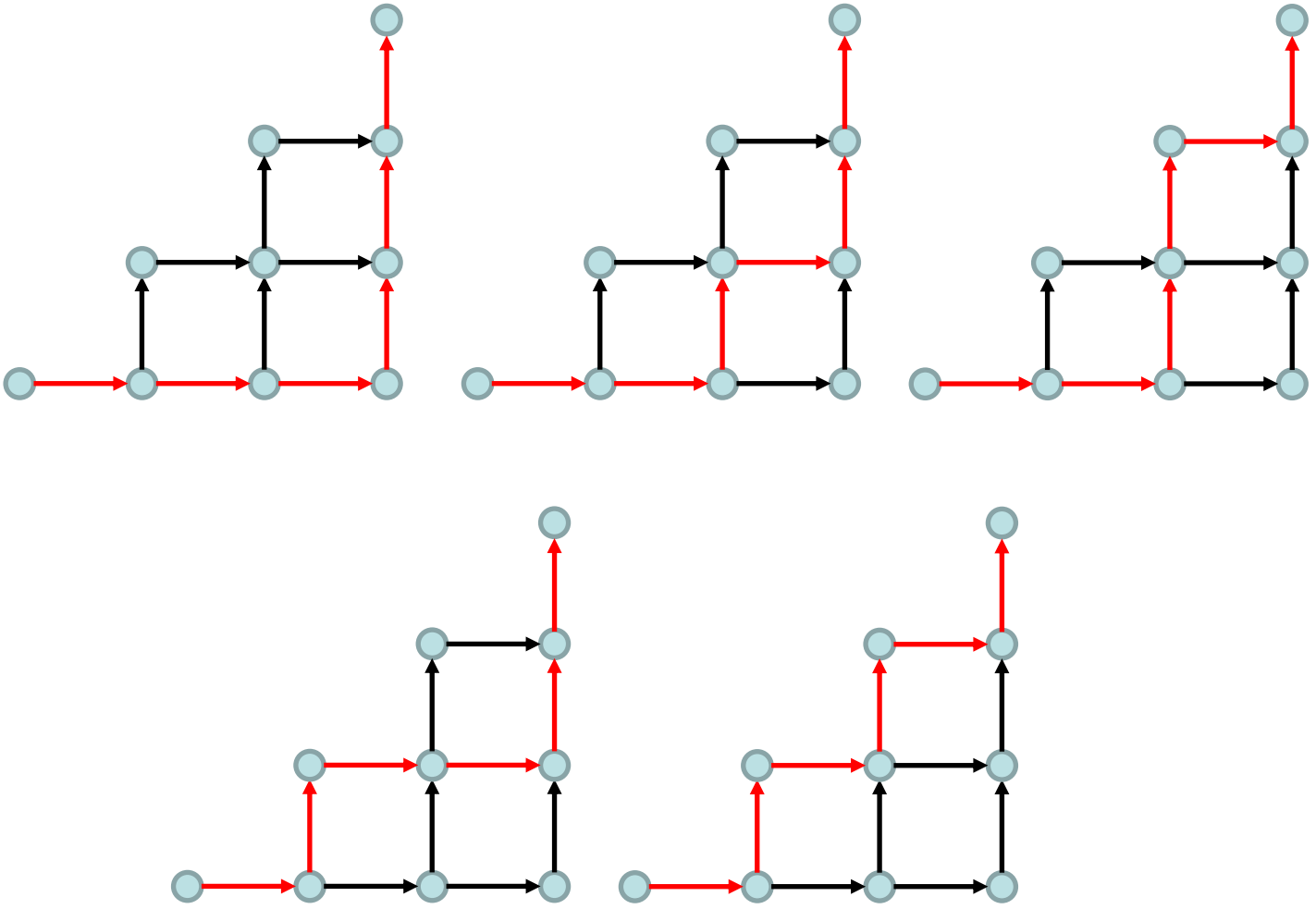
あさ → 朝
あさひ → 朝日
あし → 足
あしか → アシカ
あしかせ → 足枷
あした→ 明日
あせ → 汗
あそ → 阿蘇
いえ → 家
いか → 以下
いし → 石
うし → 牛
うしろ → 後ろ
うそ → 嘘

木構造の圧縮を考える

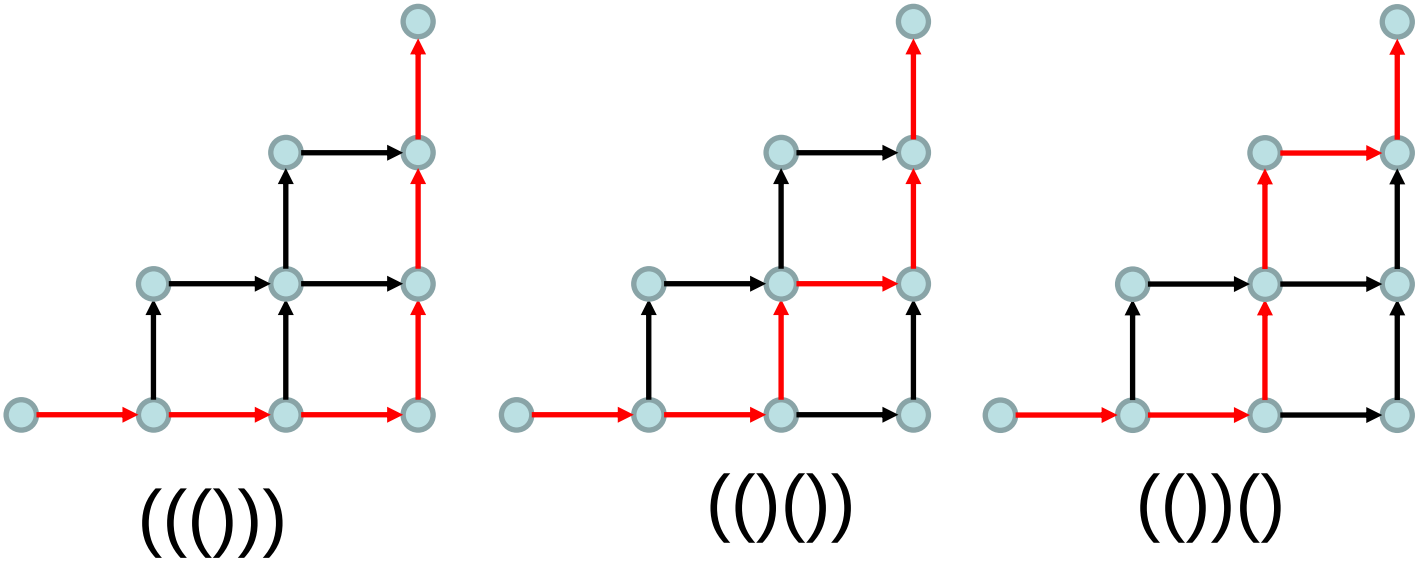
- s から t への行き方は何通り?

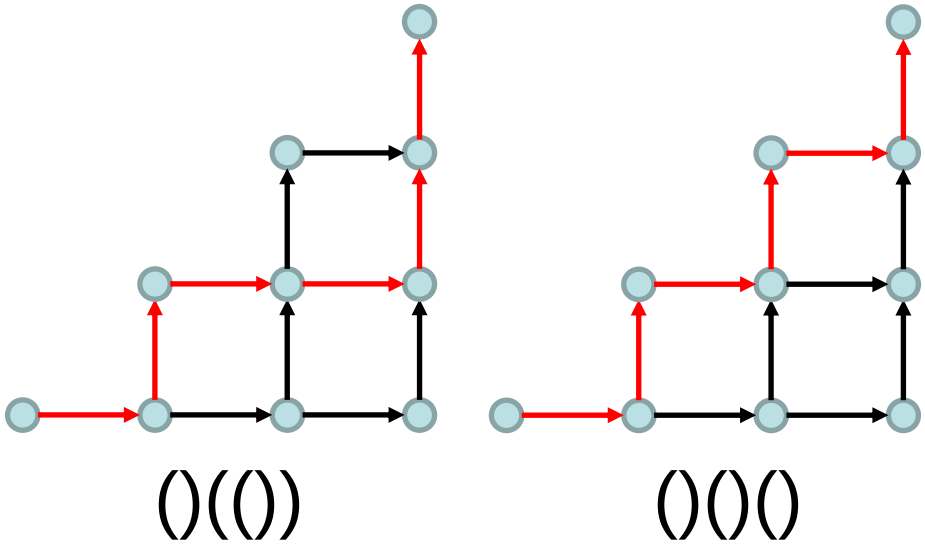


- 5通り

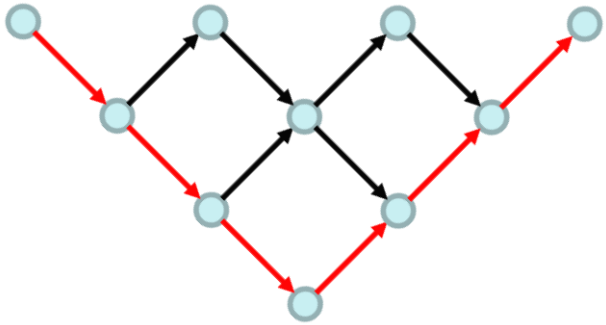


- →と (, ↑と) を対応させると，各ルートは括弧の対応の取れた(バランスした)括弧列を表す

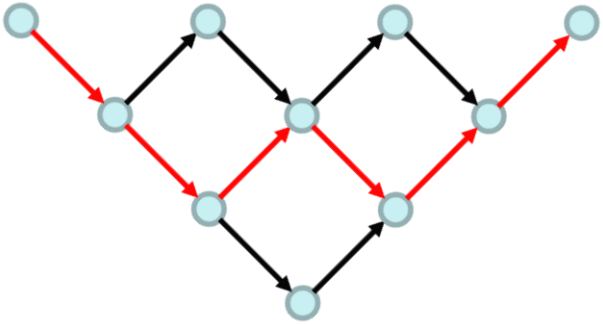




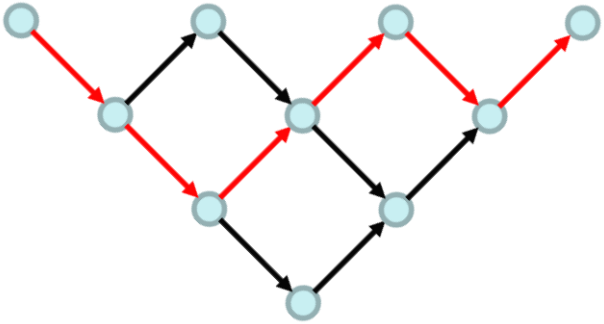
- バランスした括弧列は，深さが常に 0 以上の経路と対応する



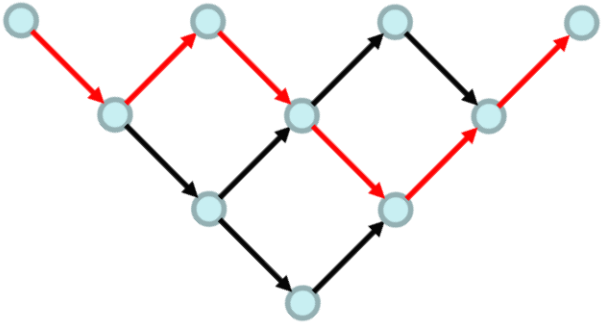
((()))



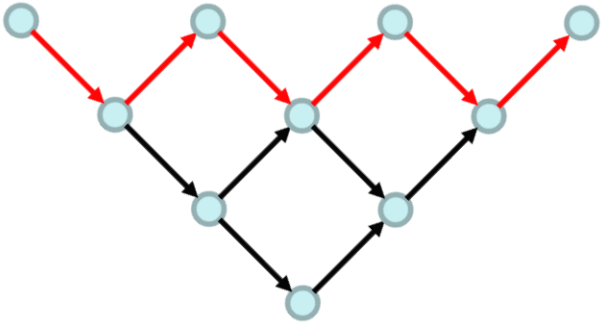
(()())



(())()

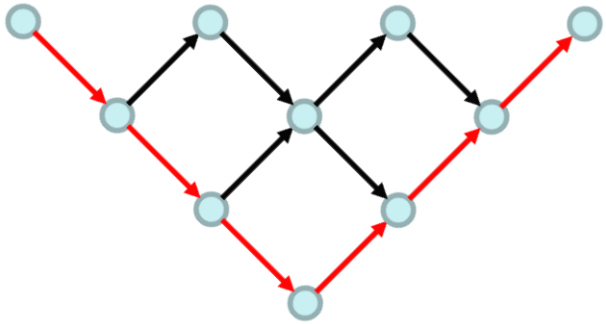


()(())

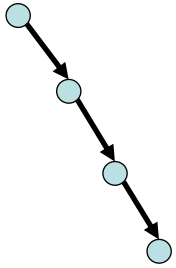


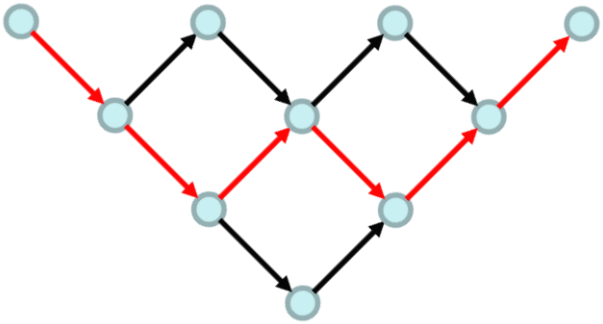
()()()

バランスした括弧列と順序木には1対1対応がある

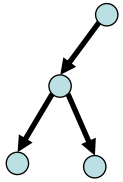


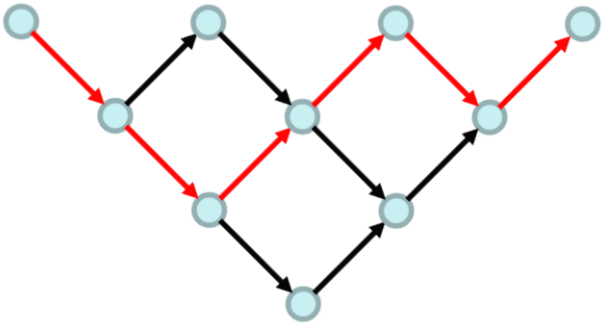
((()))



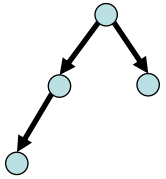


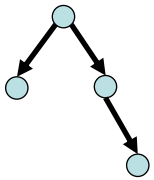
(()())

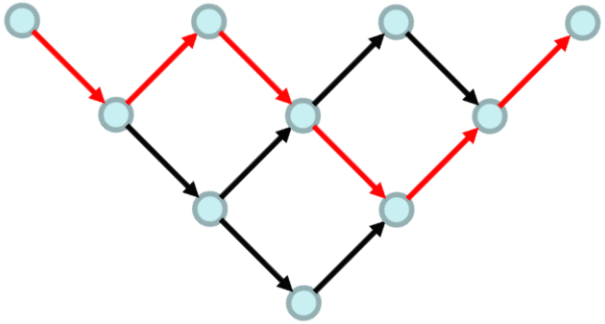




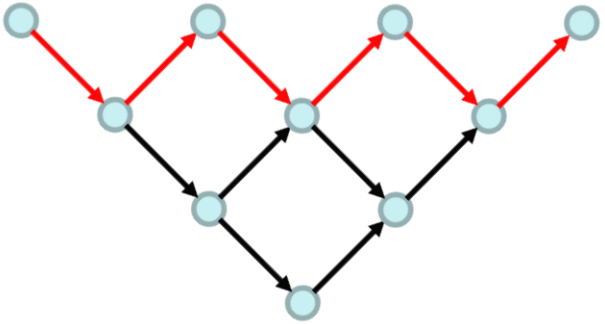
(())()



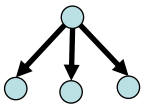




()(())

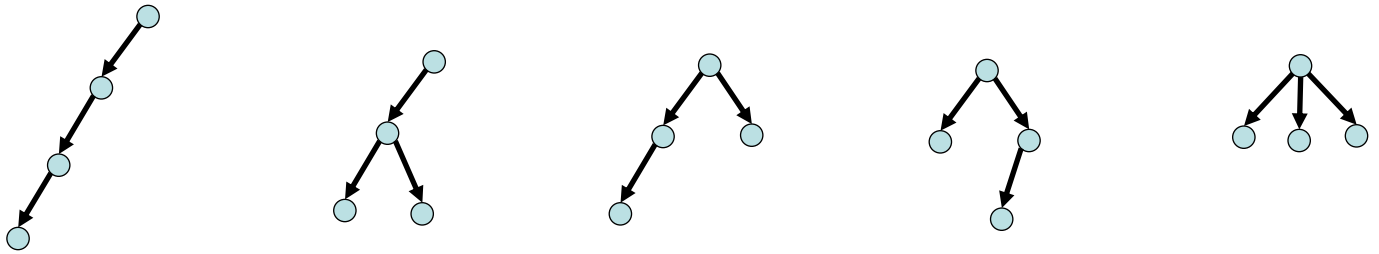


()()()



順序木の圧縮

- 木をバランスした括弧列で表現する
  - 一番外側に括弧を追加する
- n 点の木は長さ $2n$ の括弧列で表現できる
- もっと短い表現は無いか



$n = 4$

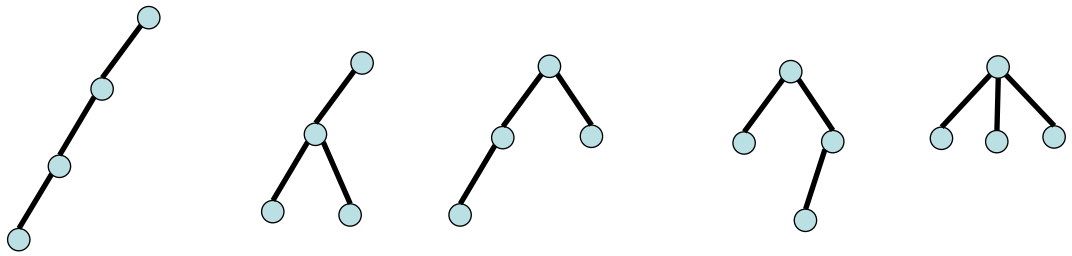
((()))　((()()))　((())())　(()(()))　(()()())

順序木の個数

- n 点の順序木の個数を T_n とする
- T_n = (長さ $2n-2$ のバランスした括弧列の数)
  < (長さ $2n-2$ の全ての括弧列の数) = 2^{2n-2}
- T_n = (縦横$n-1$マスずつの格子で対角線をまたがない経路の数)

$$T_n = \frac{1}{n}\binom{2n-2}{n-1} = C_{n-1}$$

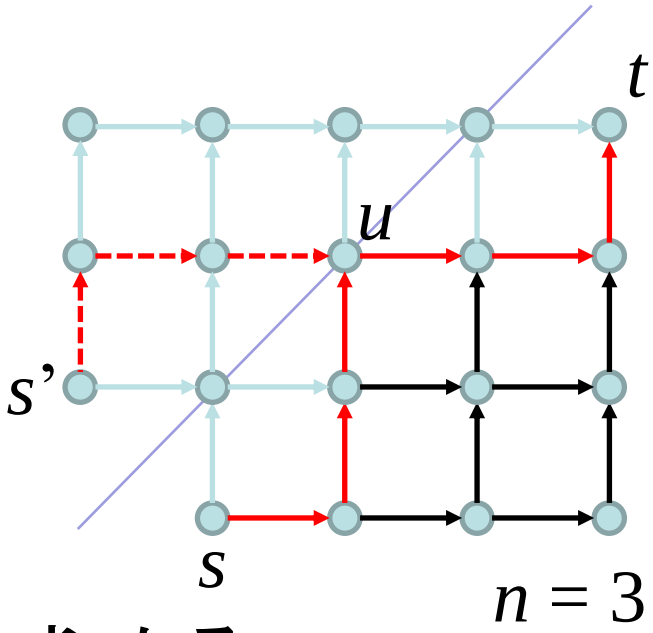
- C_n はカタラン数 (Catalan number) と呼ばれる



$n = 4$

カタラン数

$$C_n = \binom{2n}{n} - \binom{\text{対角線を跨ぐ}}{s-t\text{ 経路の数}}$$



- 対角線をまたぐ s-t 経路の数を求める
  - 対角線を初めて跨いだ点を u とする
  - s から u の経路を折り返す
  - s' から t の全ての経路の数と等しい $\binom{2n}{n-1}$

$$C_n = \binom{2n}{n} - \binom{2n}{n-1} = \frac{1}{n+1}\binom{2n}{n}$$

順序木の表現のサイズ

$$C_n = \frac{1}{n+1}\binom{2n}{n} \approx \frac{4^n}{n^{3/2}\sqrt{\pi}}$$ (スターリングの公式より)

- b ビットで表現できるものは最大 2^b 種類
- n 点の順序木は C_{n-1} 種類ある
- 順序木を表現するには $\log_2 C_n \approx 2n$ ビット必要
- つまり，括弧列による順序木の表現は
  ほぼ最適サイズ

まとめ

- 通常のデータ圧縮は，圧縮したまま処理できない
- 「場合の数」の考えを使うと，最適に圧縮でき，データを圧縮したまま使える ⇒ 簡潔データ構造
- 人のDNA配列の読み取り処理に必要なメモリ
  - 従来手法: 300GB
  - 簡潔データ構造: 3GB
- ビッグデータ処理で重要な技術

http://researchmap.jp/sada/