Computational Thinking は評価可能か：ビーバーチャレンジの協働問題解決過程の分析

Can Computational Thinking be Evaluated: Analysis of Collaborative Problem Solving Process of “Bebras”

遠山 紗矢香[1]，松澤 芳昭[2]，谷 聖一[3]
Sayaka Tohyama, Yoshiaki Matsuzawa, Seiichi Tani
[1]静岡大学，[2]青山学院大学，[3]日本大学
Shizuoka University, Aoyama Gakuin University, Nihon University
tohyama@inf.shizuoka.ac.jp

概要

本研究は，Computational Thinking (CT) とはどのような能力なのかを検討することを目指して，計算機科学と CT に関連した問題が出題される「ビーバーチャレンジ」の問題解決過程を分析した初期段階の研究である．本研究では大学生の正答率が低かった「検査」と「画像圧縮」の 2 問に焦点化して，2 名で話し合いながら問題を解かせて問題解決過程を観察した．大学生 4 ペアの分析結果を用いて，問題解決過程での CT の発現について検討した．

キーワード：ビーバーチャレンジ，協調問題解決，Computational Thinking (CT)

1. はじめに

本研究は，”Computational Thinking” (CT) とはどのような能力なのかを検討することを目指して，計算機科学と CT に関連した問題が出題される「ビーバーチャレンジ」の問題解決過程を分析した初期段階の研究である．近年では子ども向けプログラミング教育が日本の義務教育段階にも導入されただけでなく，GIGA スクール構想による児童生徒への 1 人 1 台の学習用計算機の配布など，ICT を活用した教育に関して様々な動きがある．こうした中で，CT の観点からも ICT を活用した教育の効果に対する関心が高まっていると考えられる．一方で CT が指す能力観については，現在も議論されているところである．

本研究では 2 名で話し合いながらビーバーチャレンジの問題を解く状況を設定することで，どのような知識が問題解決中に使用されるのか，どのような問題解決方略が適用されるのかといったことを観察する．この観察から得られた結果を用いて，問題解決過程で観察される知識や問題解決方略等は先行研究で指摘されてきた能力観とどのように異なるのか，また CT と呼ばれる能力観とどのように関連しているのかを検討することを，本研究の目的とする．

2. 背景

2.1. ビーバーチャレンジ

国際情報科学コンテスト「ビーバーチャレンジ」は，計算機科学と CT に関連した問題を出題することで，児童・生徒・学生が計算機科学やその関連領域に興味を持つきっかけを与えることを目指すものである[1]．このコンテストは，”Bebras“ として 2004 年にリトアニアで始まり，現在まで欧州を中心に展開が進められている．日本では 2010 年より，数理情報科学教育の裾野を広げる目的から，小学生から高校生を対象とした国際情報科学コンテストとして実施されている[2]．

ビーバーチャレンジは前述の目的に資するため，チャレンジする児童生徒がプログラミングやコンピュータの仕組みに関する特別な知識を持っていなくても回答できるように問題が作成されている．このため，長さがばらばらな色鉛筆一式を短い順に並び替える問題としてソートの問題を出題する，などの工夫がなされている．また，ブラウザでテストができるように設計されており，回答は多肢選択式である．多くの場合，選択肢は 4 つ提示され，このうち 1 つが正解である．

例年開催されている「International Bebras Task Workshop」には世界各国の協力者が参加し，新規の問題作成と検討が進められている[3]．このため，作問の際に持ち込まれる文化も多様だと考えることができる．

2.2. Computational Thinking

CT は，Wing が提唱した能力観であり[4]，日本語では「計算論的思考」と訳されている[5]．CT は，コンピュータ科学者のように考える態度や能力のことを指しており，万人にとって学びたい・使いたい普遍的なものだと主張されている．Wing[4]では，CT の具体例

として，問題を小さくすること，抽象化すること，再帰的に考えることといったものが挙げられている．

Wing の主張に続いて，CT の能力観や観点については Wing 本人が再整理を行うまでの間に様々な整理がなされた．Brennan & Resnick では，CT を 3 つの側面から整理し，それぞれと関連の深い学習者の活動を例示した[6]．Grover & Pea は，CT について 6 つの概念と 5 つのプラクティスに整理可能であることを主張した[7]．なお，Wing は 2014 年に，CT の定義について次のように説明した：“Computational thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer – human or machine – can effectively carry out” [8]．その後も CT が何を指すのかについては diSessa の提唱した “Computational Literacy” と比較されたり[9]，各学問領域における CT の解釈に幅があることを指摘されたり[10]といったことを経ながら，現在も議論は続けられている．

CT の「コンピュータ科学者のように」という考え方は，近年の子ども向けプログラミング教育の隆盛に通じているとする見方がある[9]．子ども達がプログラミングを学ぶことを通じて，コンピュータ科学者の思考習慣に触れることができるならば，Wing[8]の言う CT の育成も促される可能性がある．しかしながら，この考え方は 1980 年代の子ども向けプログラミング教育で標榜されたものに類似しており，当時の限られた一部研究でのみ成果が示されたものであるため[11]，慎重な検討が必要な点でもある．

2.3. プログラミング教育と CT

プログラミング教育がコーディングのみを教えるものではなく，CT として説明されるような能力を育成するための営みである場合，プログラミング教育の成果を CT の伸びで評価することが試みられる．計算機と CT をテーマとして作成されているビーバーチャレンジは，学習者の CT を評価する機能があると期待されるため，プログラミング教育の評価として使用した先行研究がある．

Matsuzawa ら[12]は，プログラミングの実技テストやペーパーテストとビーバーチャレンジの結果に一定の相関関係が見られたことを示している．一方で Djambong ら[13]や Dolgopolovas ら[14]の研究では，学習者の授業等での成果とビーバーチャレンジの得点に相関関係が見られなかったことが示されている．Dolgopolovas らはこの結果について，調査対象であった学生が受講していたプログラミング授業の改善が必要であるという考察を行っている．つまり Dolgopolovas らは，受講生の達成度がビーバーチャレンジの結果として反映されることを期待していた．この期待は，授業内容と，出題したビーバーチャレンジの一部問題との間に類似性が見られたことから生じていると考えられる．

プログラミング教育とビーバーチャレンジの関係性について，先行研究によって異なる結果が示されていることについて検討するためには，CT が指す幅広い能力観の中で，ビーバーチャレンジが CT のどのような側面をテストしているのかを調査する必要があると考えられる．そこで本研究では，ビーバーチャレンジで出題されている問題が，問題解決者のどのような知識や考え方を駆動するものであるかを検討することを目的とする．

3. 研究方法

3.1. 問題の概要

本研究では，ビーバーチャレンジの中で高校生に向けて作成されたシニア問題の中でも「検査[1]」と「画像圧縮[2]」に焦点化して，2 名で話し合いながら問題を解かせることで，問題解決過程を観察する．2 つの問題を本稿末尾の付録に示す．これらの問題は，2019 年度にプログラミング言語 Java を用いてプログラミングを半年間学んだ文系学科に所属する大学学部 1 年生 60 名が取り組んだにもかかわらず，正答率が 50%に満たなかった問題であったため，一定の難易度が認められると考え，本研究で採用した．

いずれの問題も，問題解決上の決まりや，行わなければならないことは説明文の形で問題に全て示されており，事前知識は不要である．ただし，「検査」はプログラミング言語 BASIC のように行番号を指定して処理を行うプログラムや go to 文が使用できるプログラミング言語でのプログラミングを経験している場合，「画像圧縮」は再帰的な考えに触れたことがあったり「分割統治」といったアルゴリズムを学んだことがあ

[1] http://bebras.eplang.jp/index.php?2018-検査
[2] http://bebras.eplang.jp/index.php?2017-画像圧縮

ったりする場合には，これらの経験を応用できるため，正答しやすい可能性がある．問題の特徴を次節に示す．

3.2. 「検査」問題の特徴

「検査」は，自然言語で書かれた条件分岐や変数を含むプログラムを解読して実行結果を正確に予想する問題である．各行には番号が振られており，いわゆる「go to 文」と呼ばれる処理が埋め込まれている．手続き型のプログラムは一般的に上から順に処理されるが，go to 文では無条件で指定された行番号へと処理が移る[3]．したがって検査の問題を解くことで go to 文の仕組みに触れることができる．

検査の問題は，自然言語で示されたプログラムを頭の中で実行すると，反復処理の中で変数の値が増加していくことに気付く．また，go to 文の働きによってプログラムの特定部分が繰り返し無限に実行される形式となっているため，変数の値は無限に増加する．しかしながら「変数が示す値の分だけ容器を振る」というプログラムが実行されることはないため，増加していく変数は参照されないままプログラムが終了する．

3.3. 「画像圧縮」問題の特徴

「画像圧縮」は，8×8 マスの二次元平面の各マスに 1 か 0 の値が設定された図が与えられ，その図に示された情報に対して与えられた手続きを適用することで，図の情報量を少なくするものである．各マスの値を全て記憶しようとすると情報量が多くなるが，隣り合うマスの値が同じ限りは範囲と値を記録しておけば，記憶すべき情報量を少なくできる場合がある．この問題では，再帰的なデータ構造を用いた画像圧縮アルゴリズムを体験することを主眼としている．

3.4. 調査方法

プログラミングを約 1 年間学んだ国立 A 大学の 2 年生に対して，「検査」と「画像圧縮」の問題を解かせた．各問題につきのべ 4 ペア，8 名分の回答を得た．このうち，2 ペアはいわゆる文系の学科に所属する学生であり，残り 2 ペアはいわゆる理系の学科に所属する学生であった．文系学科の学生と理系学科の学生はいずれも，プログラミングの基本的な概念や操作を学ぶ授業を 1 年次に履修していた．ただし，授業は文系と理系で別々に実施されていた．

参加した大学生らのプログラミング授業での到達度はいずれも，クラス全体の中程度以上であった．中程度の到達度であれば，学習者が 1 人で，基本的な制御構造を数個組み合わせて 10〜20 行程度のプログラムを困難なく完成させることができる．実験協力者の学生が用いてきたのはプログラミング言語 Java であった．

研究の手続きは遠山・白水[15]を参考にし，1 人で問題を解いた後で，2 人で同じ問題を再度解くデザインとした．ただし，本調査では，一部参加者において，遠山・白水[15]で行われていた以下の 3 までのステップでは未消化感が観察されることがあったため，参加者の希望に応じて 4 のステップを追加した．なお，1 問あたりの時間を 3 分としたのは，ビーバーチャレンジで設定されていた 1 問あたりの標準問題解決時間が 3 分程度であったためである．

1. 1 人で問題を解く（3 分）
2. 1 人で解いたものと同じ問題を今度は 2 人で話し合いながら解く（3 分）
3. 1 人で同じ問題の解き方を実験者へ説明する（3 分程度）
4. 再度 2 人で納得できるまで同じ問題を解く（時間は任意）

3.5. 分析方法

ステップ 1〜4 について，各実験協力者の各問題の回答の正誤を分析した．また，ステップ 2 と 4 については，発話の書き起こしを行った上で，書き起こした発話について分析を行った．なお，ステップ 3 はステップ 1，2，4 の回答を解釈するための補足資料として使用した．

4. 結果

4.1. 回答の正誤

以下では便宜的に，本研究の実験協力者ペアを文系ペア（文系学科の学生 2 名）・理系ペア（理系学科の学生 2 名）と呼ぶ．ステップ 1〜4 の個人回答およびペア回答は表 1 の結果となった．表では，ステップ 1 の回答を左に，以後順にステップ 4 の回答までを示している．○は正答，×は誤答（どの選択肢も選んでいない

[3] go to 文は近年のプログラミング言語では使用が推奨されていないことが多い．例えばプログラミング言語 Java では go to 文が実装されていない．

未回答状態を含む）であり，書き間違いやケアレスミスのみで考え方が合っていれば正答とした．理系ペア 1 を除く 4 ペアでは，少なくとも 1 つの問題で，2 名で話し合うことによって 1 名のときよりも回答の質が向上していた．

一方で理系ペア 1 は，2 名で話し合っても回答の質が 1 名の時と同様あるいは質が下がっていた．このペアは検査の問題において 1 名が，出題者が意図していなかった方法で問題文を読み解いたために誤答していた．残りの 1 名は，ステップ 1 では出題者の意図通りに問題文を読解していたが，ステップ 2 の話し合いで相手の問題文の読み取り方に賛同したことで，ペアの回答としては誤答となっていた．

表 1　ステップ 1〜4 の各回答の正誤

	検査	画像圧縮
文系ペア 1	○○→○→○○	
文系ペア 2		××→×→××→○
文系ペア 3	××→○→○○→○	××→○→○○→○
理系ペア 1	○×→×→××→×	○○→○→○○→○
理系ペア 2	○○→○→○○→○	××→○→○○→○

4.2. 発話分析

ステップ 2 の問題解決における対話で観察された実験参加者の発話のうち，特徴的だったものを要約して 4.2.1 項と 4.2.2 項に示す．「検査」については，出題者が問おうとしたと考えられる，変数と容器を振る回数の区別が付いているかどうかという点だけでなく，問題文に示された手続きが多様に解釈できる可能性があることが，理系ペア 1 の発話から示された．「画像圧縮」については，与えられた図をどのように切り分けてどの部分から処理をしていくのかについて，2 人で問題を読みながら答えを創り上げていく過程が見られた（文系ペア 2，文系ペア 3，理系ペア 2）．

4.2.1. 検査の問題

- 文系ペア 1：行番号の数字と，変数に加える数字とが示されており，どちらがどちらを指しているのか途中で混乱した．
- 文系ペア 3：プログラムの中で変数の値がどんどん増えていくので，容器を振る回数もどんどん増えていくと勘違いしていたが，話し合いによってそうではないことに気付いた．
- 理系ペア 1：プログラムがループして終わらないので「おかしい」と思った．なので「go to X という命令文を含んでいれば，装置は次にプログラムの行 X を読み込み，実行を継続します」という問題文の部分を考え直した．その結果，「実行を継続します」という説明は，go to 文で指定された X 行目のプログラムだけを実行したら，その後は X+1 行目へ処理が移るのではなく，go to 文が書かれていたすぐ次の行に処理が移るという意味だと考えた．
- 理系ペア 2：「A 回容器を振る」というプログラムが書かれている行が，プログラムのどこからも呼び出されていないため，容器が振られることはない．

4.2.2. 画像圧縮の問題

- 文系ペア 2：「4 つの並びに分ける」という説明の意味を誤解していた．（4×4 ではなく，2×4 だと誤解していた）
- 文系ペア 3：「4 つの並びに分ける」という言葉の意味や，どのマスから処理を始めるのかがわからなかった．
- 理系ペア 1：迷いなく解くことができた．ただし，どのマスから処理を始めるのかを示している矢印の図が本文の説明と違ったため混乱した．
- 理系ペア 2：説明がどのような手続きを表しているのかうまく読み取れなかったため，話し合いながら少しずつ手続きを確認した．またどのマスから処理を始めるのかを示している矢印の図が本文の説明と違ったため混乱した．

5. 考察と今後の展望

本研究では，ビーバーチャレンジの問題解決過程でどのような知識や問題解決方略が使用されているのかを観察するため，2 種類の異なる問題について各 4 ペアの問題解決過程を分析した．その結果，検査の問題では，作問者が意図的に盛り込んだと考えられる，変数の値が増加していくことと「容器を○回振る」という処理とを混同しがちであった点に加えて，行番号と変数の値の区別がつき難いことや,「実行を継続します」という問題文が多様に解釈できることが，問題解決を困難にしていたことが示された．

また，画像圧縮の問題では，問題文が示している内容がひとたび解釈できれば正答できる傾向が示されたものの,「4 つの並びに分ける」という説明が何を指しているのかを読み取ったり，画像と問題文の説明が異なっているように見えたりしたことが問題解決を難しくしていたことが示された．上記の点は．CT が指している能力を回答者が発揮するよりも前の段階で，問題解決に支障をきたしていたようにも捉えられる．

2 つの問題を比較すると，検査の問題では，問題文で示されたルールをそのままプログラムの読み取りに適用することが求められていたのに対して，画像圧縮の問題では，問題文に示された手続きを正確に読み取ることだけでなく，その手続きを自分が使って問題を解くことが求められていた．その意味では，これら 2 つの問題はどちらも難易度が高い問題ではあったが，後者はより知識の活用が求められる問題だった可能性がある．

理系と文系の学生を比較すると，どちらの学生も 2 人で話し合うことで問題を解決できていたという意味では，大きな差はなかった．ただし，理系学生が検査の問題文について考えこんだことは当初予想されていなかった. 問題文の先頭で「容器を定期的に振ります」と書かれているにもかかわらず，与えられたプログラムが無限ループに陥っている点についても，理系学生は混乱していた．このため，問題の文脈設定にも更なる配慮が必要だと考えられる．

1 人と 2 人の問題解決過程を比較すると，検査の問題では，プログラムの挙動を正確に予測するうえで 2 人の話し合いが役立っていた可能性が，1 ペアの結果から示された．画像圧縮の問題では，3 ペアの結果から，求められている手続きを解釈するうえで 2 人での話し合いが有用だった可能性が示された．これらの話し合いが, CT 発現よりも前の段階での問題解決を支援しただけなのか，それとも CT を用いて問題解決を行う過程を支援していたのか, 今後検討する必要がある．

本研究では，今後の方向性について検討するため，異なる傾向を持つ少数の対象者について調査を行った事例について報告した．今後は引き続き，一部改善した問題を用いるなどしながら，検討を進めていきたい．

文献

[1] Bebras. https://www.bebras.org (2021.07.06 参照)

[2] ビーバーチャレンジ. https://www.ioi-jp.org/junior/bebras2020.html (2021.07.06 参照)

[3] 谷聖一・兼宗進・中野由章 (2011). “国際情報科学コンテスト Bebras の問題を検討する Bebras Workshop 参加報告”, 情報処理学会研究報告, Vol.2011-CE-111 No.7, pp. 1-5.

[4] Wing, J. (2006). “Computational thinking”. *Communications of the ACM*, 49(3), 33–35.

[5] 中島秀之. (2015). “計算論的思考”, 情報処理, 56(6), pp. 584-587.

[6] Brennan, K., & Resnick, M. (2012). ”New frameworks for studying and assessing the development of computational thinking”. In Annual American Educational Research Association meeting, Vancouver, BC, Canada.

[7] Grover, S., & Pea, R. (2013). “Computational thinking in K-12: A review of the state of the field”. *Educational Researcher*, 42(1), 38–43.

[8] Wing, J (2014). “Computational Thinking Benefits Society”. Social Issues in Computing. http://socialissues.cs.toronto.edu/2014/01/computational-thinking/ (: 2021/07/06)

[9] diSessa, A. (2017). “Computational Literacy and “The Big Picture” Concerning Computers in Mathematics Education”, *Mathematical Thinking and Learning*, 20(1), 3-31.

[10] Tohyama, S., Matsuzawa, Y., Yokoyama, S., Koguchi, T. & Takeuchi, Y. (2017). “Constructive Interaction on Collaborative Programming: Case Study for Grade 6 Students Group”. In *Tomorrow’s Learning: Involving Everyone – Learning with and about Technologies and Computing (IFIP AICT 515)*, pp.589-598, Springer.

[11] Li, Y., Schoenfeld A. H., diSessa A. A., Graesser A. C., Benson, L. C., English, L. D. & Duschl, R. A. (2020). “Computational Thinking Is More about Thinking than Computing”. *Journal for STEM Education Research*, 3,1-18.

[12] Matsuzawa, Y., Murata, K., & Tani, S. (2019). “Multivocal Challenge Toward Measuring Computational Thinking Bebras Challenge Versus Computer Programming”. In D. Passey *et al.* (Eds.): *OCCE 2018, IFIP AICT 524*, pp. 1–11.

[13] Djambong, T., Freiman, V. (2016). “Task-based assessment of students’ computational thinking skills developed through visual programming or tangible coding environments”. In: Sampson, D.G., Spector, J.M., Ifenthaler, D., Isaias, P. (Eds.) *Proceedings of the International Conference on Cognition and Exploratory Learning in the Digital Age (CELDA)*, pp. 41–51.

[14] Dolgopolovas, V., Jevsikova, T., Savulionienė, L., Dagienė, V. (2015) “On Evaluation of computational

think- ing of software engineering novice students". *Proceedings of the IFIP TC3 Working Conference "A New Culture of Learning: Computing and next Generations*, 90–99.
[15] 遠山紗矢香・白水始（2017）．"協調的問題解決能力をいかに評価するかー協調問題解決過程の対話データを用いた横断分析ー". 認知科学，24(4), 494-517.

＜付録＞

【検査】

ある臨床検査用の検査装置は，患者から採取した標本を入れた容器を定期的に振ります。

その装置はコンピュータプログラムで制御されていて，そのプログラムは行番号付きで記述されます。装置はプログラムを 1 行ずつ読み込みます．1 行読み込むとその行を直ちに実行します。もし，ある行が "go to X" という命令を含んでいれば，装置は次にプログラムの行 X を読み込み，実行を継続します。

プログラムは，数を A に記憶したり，A に記憶されている数に 1 を加えたり，A に記憶されている数を他の数と比較したりできます。

プログラム:

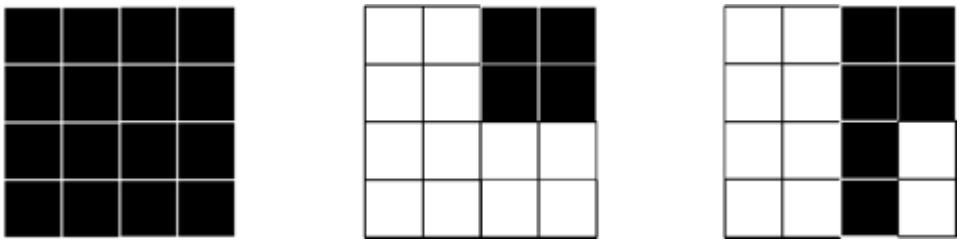
1. A に 0 を記憶
2. A に 1 を加える
3. go to 6
4. もし A が 60 と等しければ go to 8
5. A に 0 を記憶
6. A に 1 を加える
7. go to 2
8. A 回容器を振る
9. end

このプログラムを実行すると標本を入れた容器は何回振られるでしょう？

容器は振られることはない
容器はちょうど 1 回振られる
容器は 60 回振られる
このプログラムは容器を振り出すと止まらない

【画像圧縮】

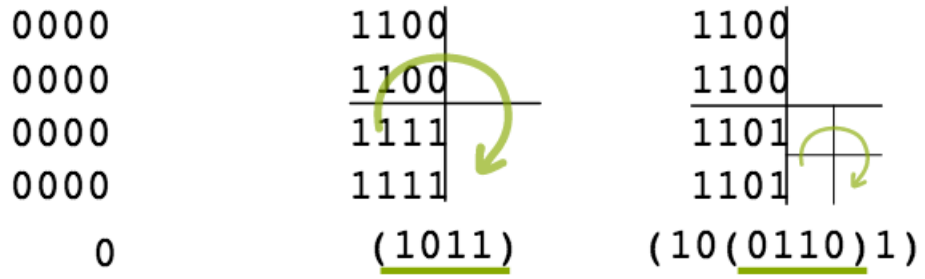
下の縦横 4 ピクセルの白黒画像を見てみましょう。



このような画像は，「白は 1」「黒は 0」と 2 種類の数字で表せます。

縦横 4 ピクセルの画像を表すには 16 個の数字を使いますが，次のような画像圧縮方法を使うと，特に単純な形

の場合には，少ない数字で画像を表せます。



まず，画像を表すピクセルと同じように，0 と 1 の数字を縦横に並べます。
縦横の 0 と 1 の並びに，次のようにこの圧縮方法を施し，結果の数字の並びを作ります。

手順 1
　縦横の数字がすべて 0 のとき，結果は 0。
　縦横の数字がすべて 1 のとき，結果は 1。
手順 2
　そうでない場合は，そうでない場合は，並びを縦と横に半分ずつにして 4 つの並びに分けます。そして，分けられたそれぞれの並びに左上から時計回りに，この圧縮方法を施します。それぞれの圧縮結果の数字の並びは「(」と「)」で囲み左から並べて書きます。真ん中の図と右の図はこのやり方を説明しています。
　分けられた並びは，1 個の数字だけのこともあります。右の図の右下の並びでは，4 つに分けられたそれぞれには手順 1 だけを使います。

```
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 0 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
```

上の図は、ある縦横 8 ピクセルの白黒画像に対する 0 と 1 の並びです。
これに圧縮方法を施して得られるのはどれでしょう？

(11(1011)1))
(111(1(1011)11))
(1110)
(111(1(1101)11))