

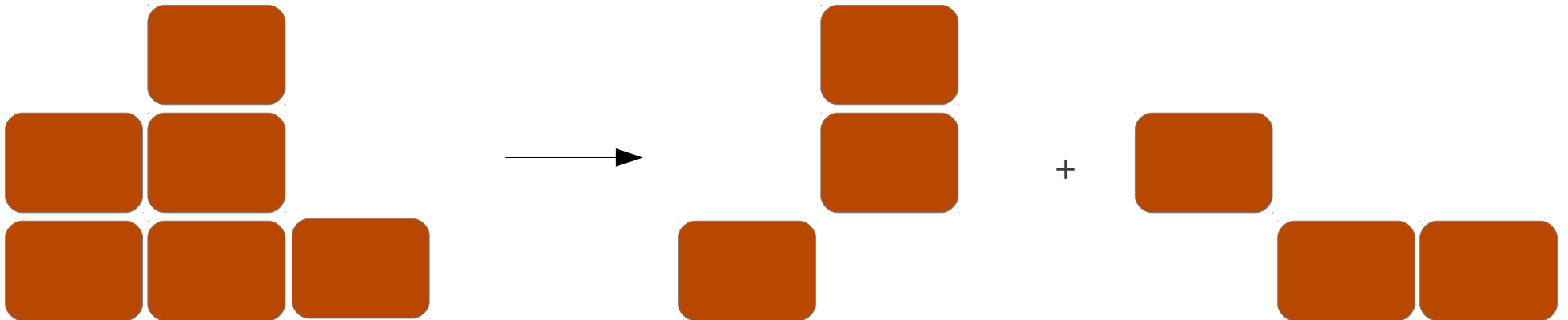
Intelligent Congruent Partition of Chocolate

2012/05/25

ルール

- チョコを 2 つに分割しよう！
 - 合同になるように
 - 回転、反転は OK
 - 斜めはダメ
 - つながってなくちゃダメ

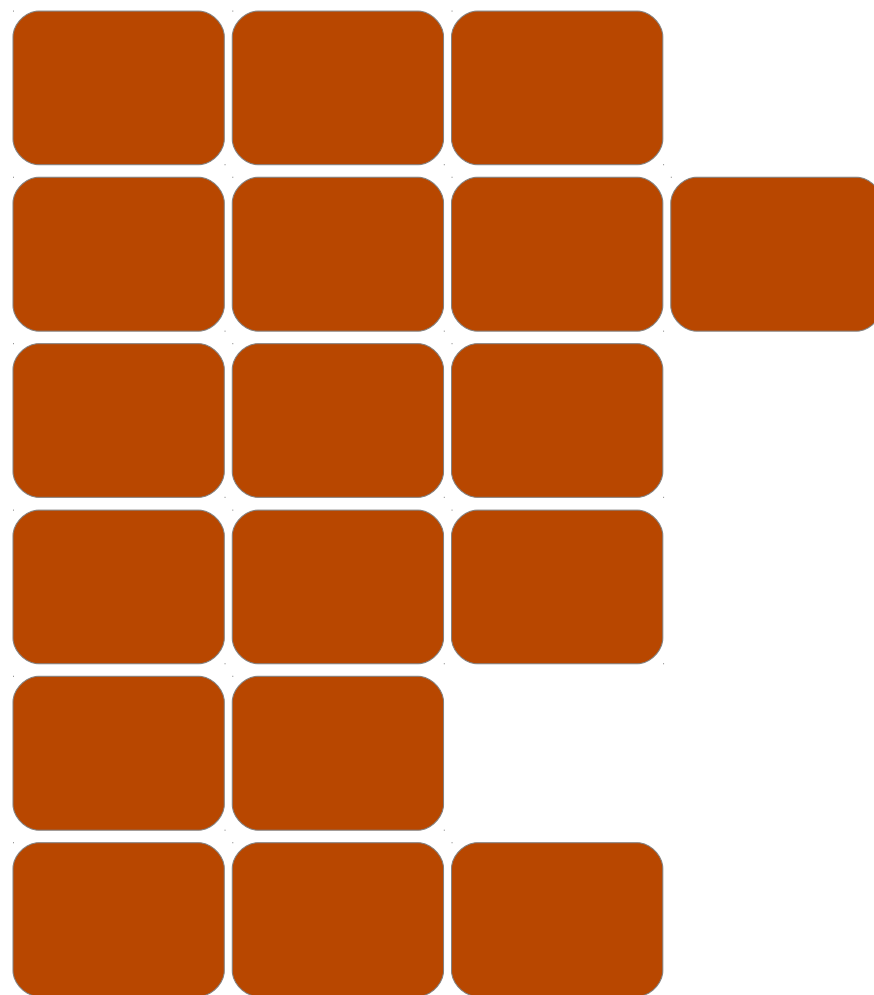
ダメな例



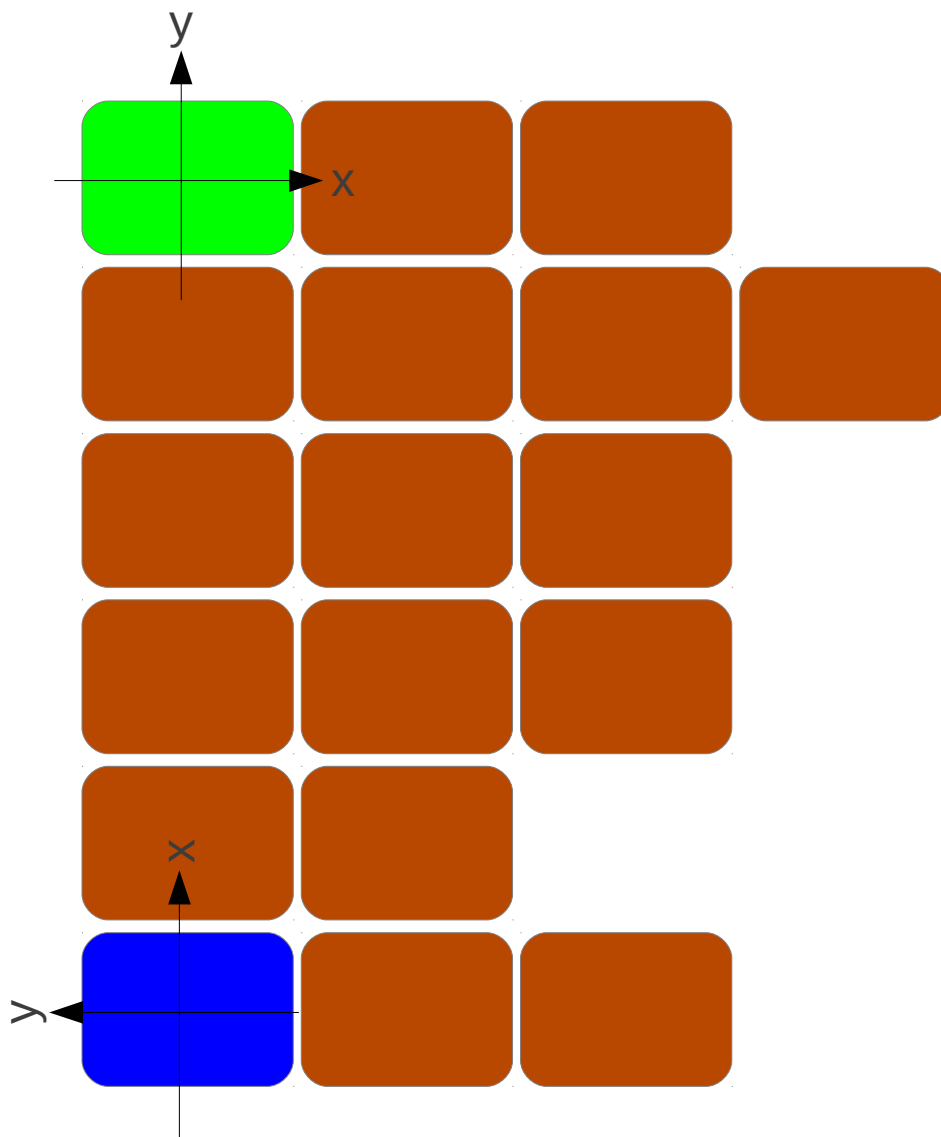
方針

- A と B の基準点を決める
 - A に 1 つのピース (P) を加える
 - $A \rightarrow P = B \rightarrow P'$ となるような P' を B に加える

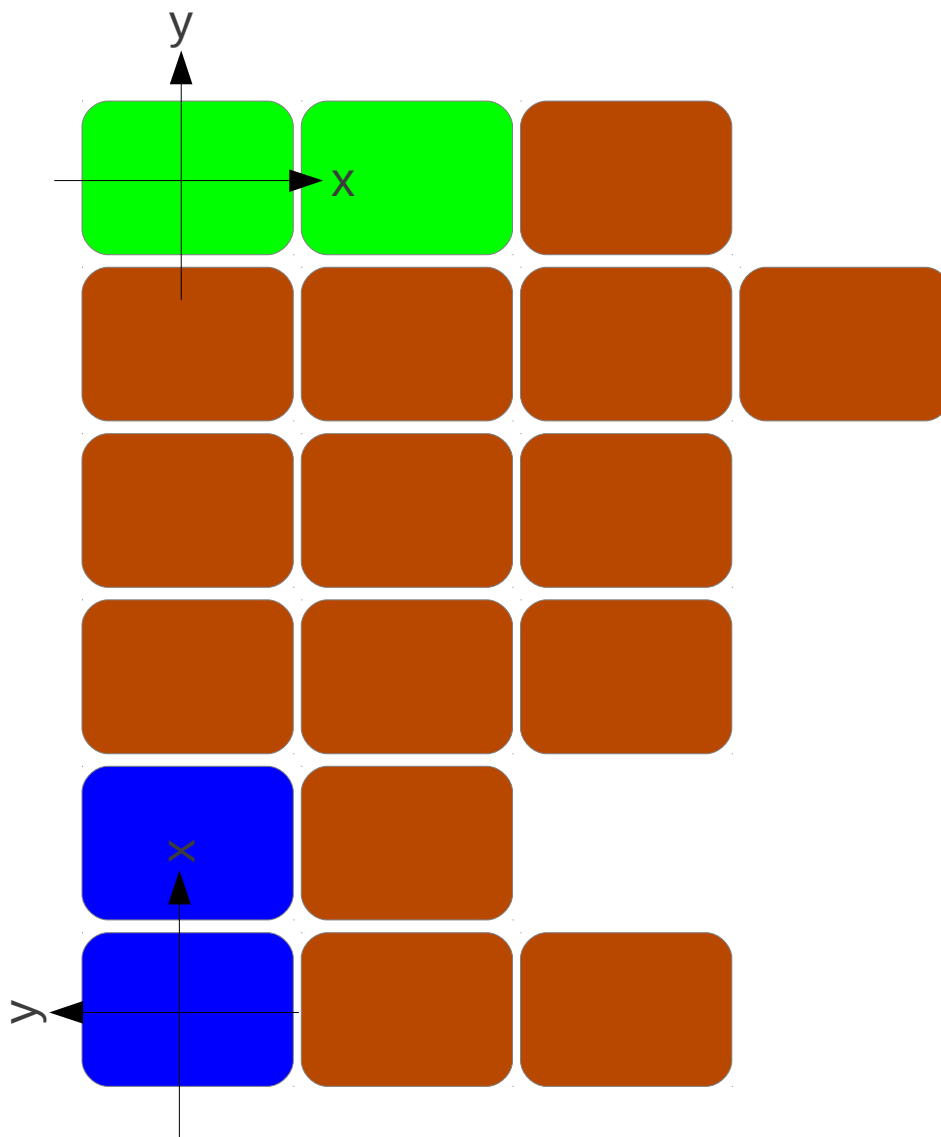
まずは感覚的に



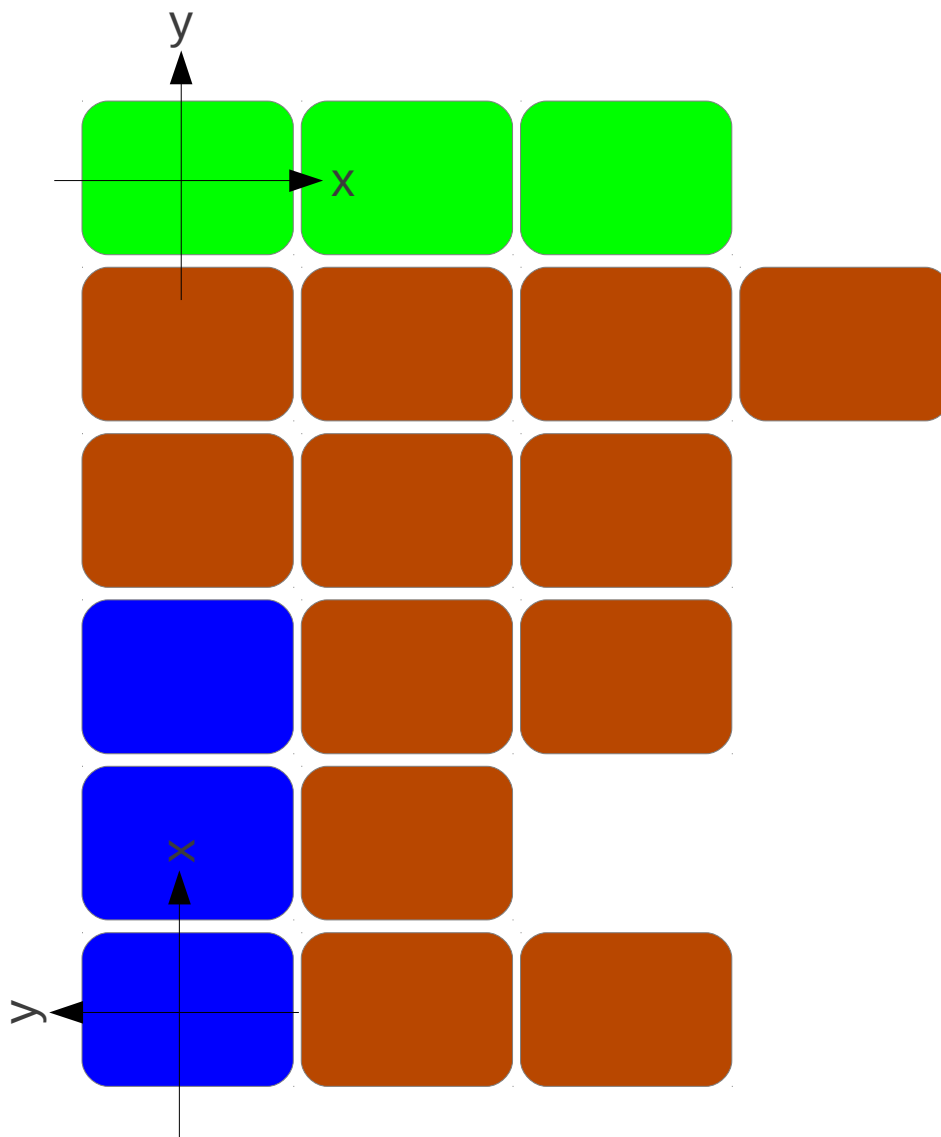
$(0, 0)$



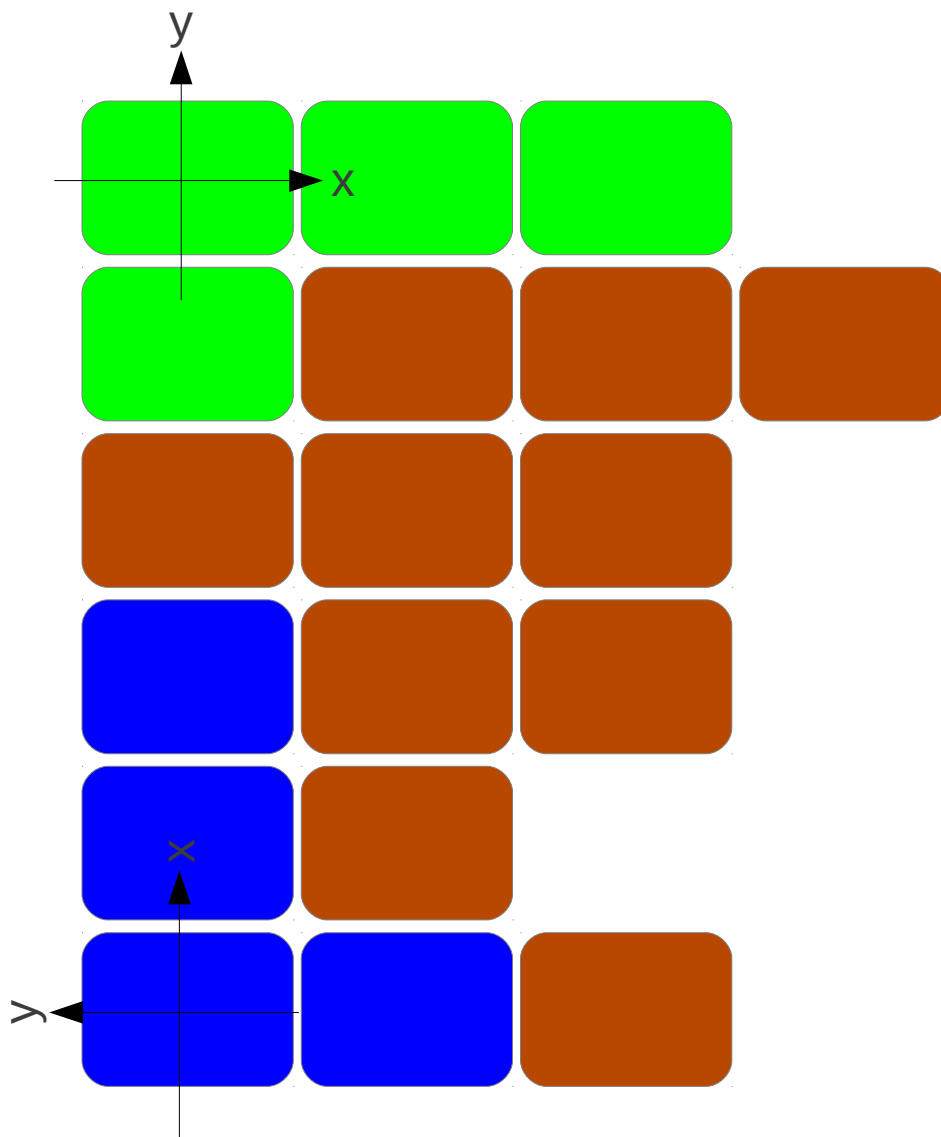
$(1, 0)$

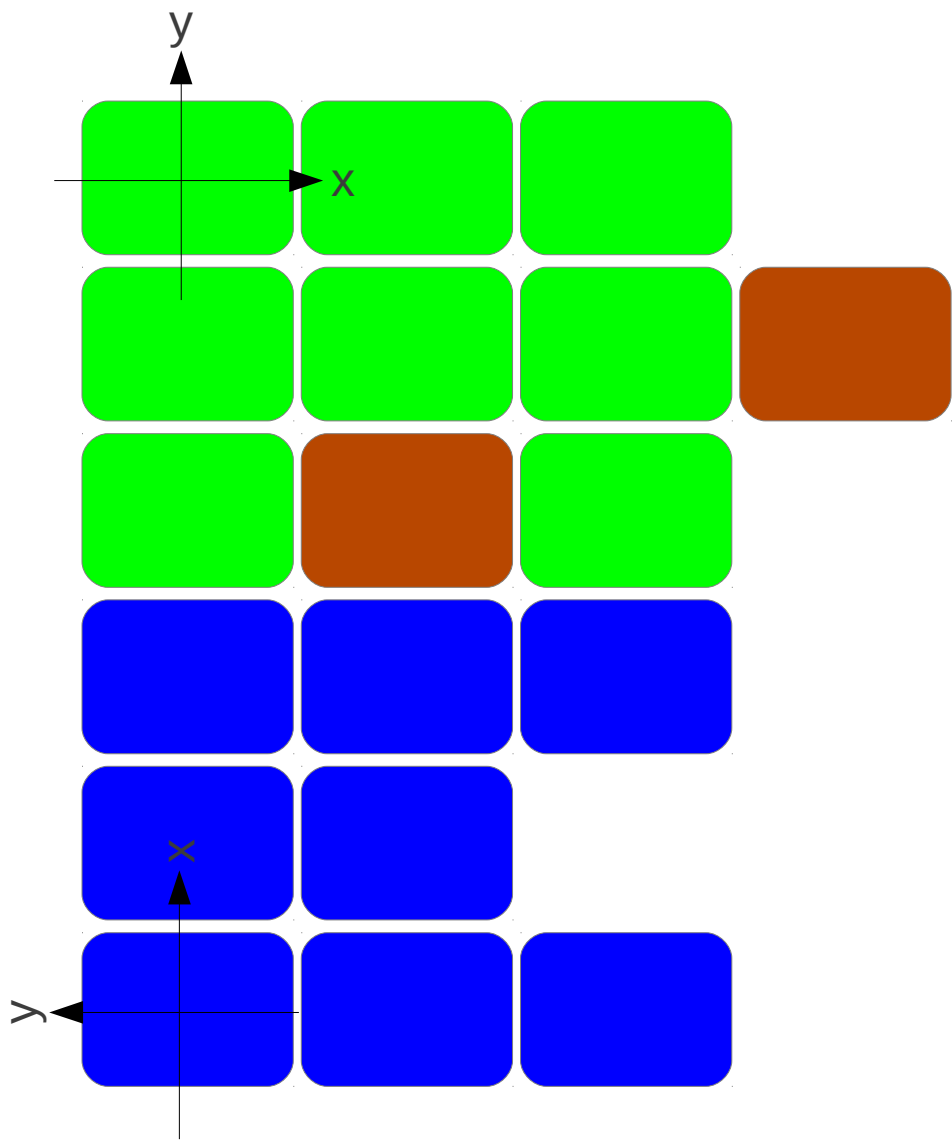


$(2, 0)$



$(0, -1)$





方針

- 固定するもの
 - Aの基準点
 - Aの座標系
- 変化させるもの
 - Bの基準点 (N通り)
 - Bの座標系 (回転 4 X 反転 2)

計算量

- B の基準点 N 通り
- B の座標系 8 通り

計算量

- B の基準点 N 通り
- B の座標系 8 通り
- ピース追加処理
 - 隣接するまだ追加されていないピースを追加
→ 最大 3 通り / 1 追加処理 $\Rightarrow 3^{(N/2 - 1)}$
 - 追加処理は深さ優先探索

計算量

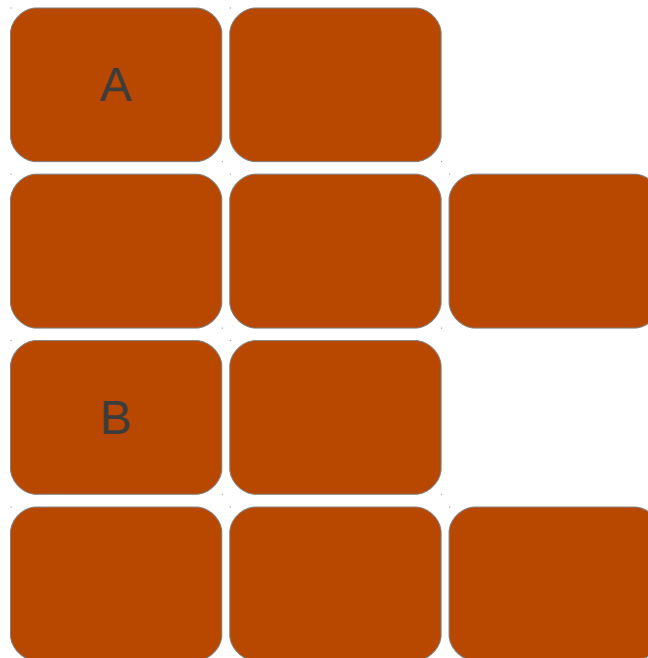
- B の基準点 N 通り $O(8N * 3^{(N/2 - 1)})$
- B の座標系 8 通り $288 * 3^{(17)}$
 $= 37192366944$
- ピース追加処理 cf. $10! = 3628800$
 - 隣接するまだ追加されていないピースを追加
→ 最大 3 通り / 1 追加処理 $\Rightarrow 3^{(N/2 - 1)}$
 - 追加処理は深さ優先探索

実装

- Phase 1
 - 対応関係を調べる
 - 対応関係がないものを枝刈り
- Phase 2
 - バックトラッキング
 - Aに隣接する1ピースを追加する
 - 対応するようにBにピースを追加する

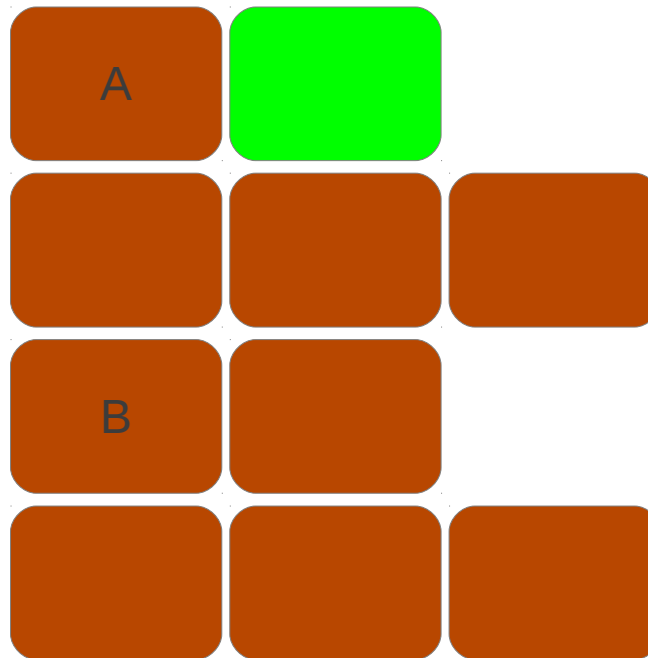
Phase 1. 対応関係

- A の基準、 B の基準を決めた



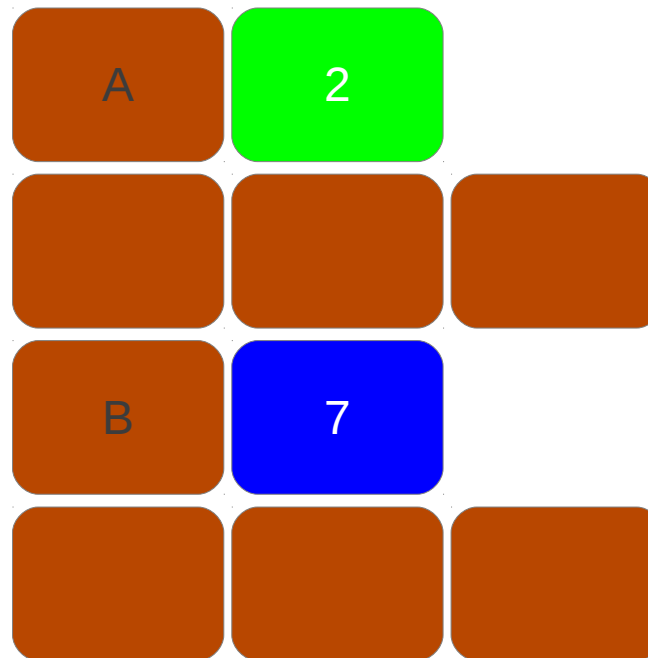
Phase 1. 対応関係

- それぞれのピースに対して



Phase 1. 対応関係

- それぞれのピースに対して

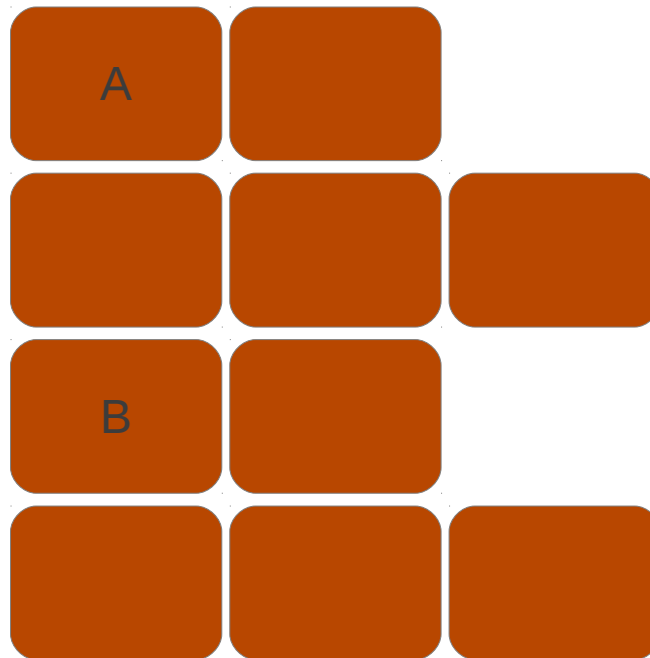


対応あり

pair[2] = 7
rpair[7] = 2

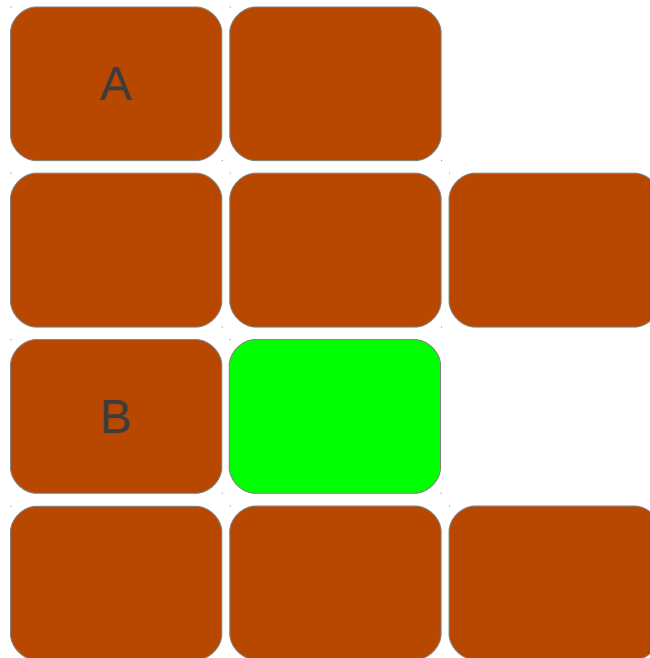
Phase 1. 対応関係

- それぞれのピースに対して



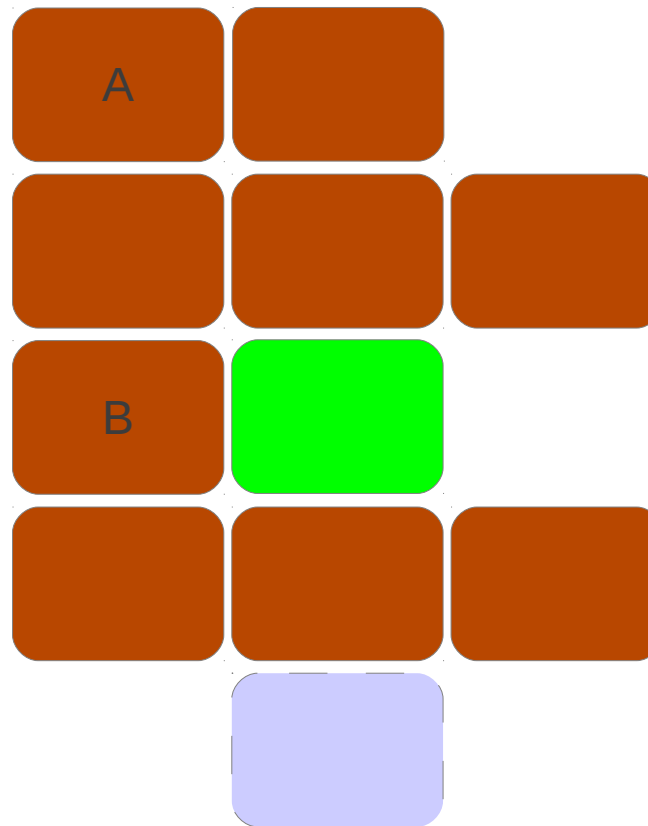
Phase 1. 対応関係

- それぞれのピースに対して



Phase 1. 対応関係

- それぞれのピースに対して



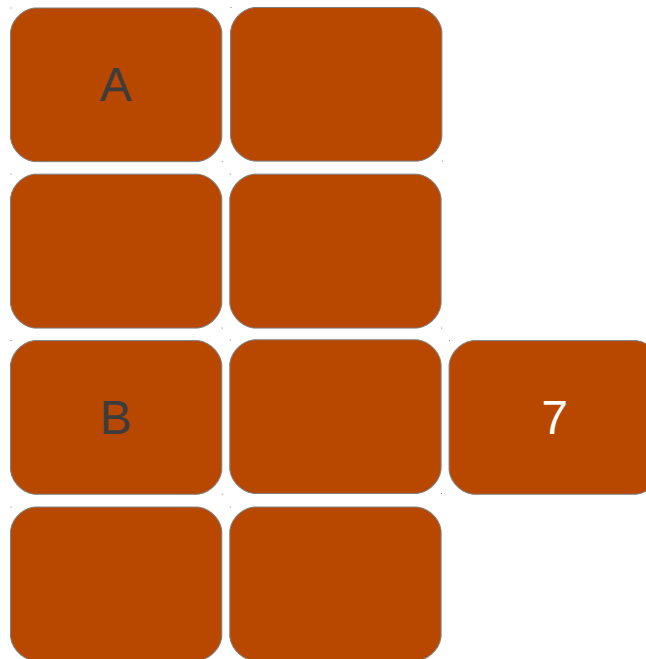
対応なし

Phase 1. 対応関係

- 枝刈りする場合
 - $A \rightarrow P$ への $B \rightarrow P'$ が存在せず、
 - $B \rightarrow P$ への $A \rightarrow P'$ も存在しない場合

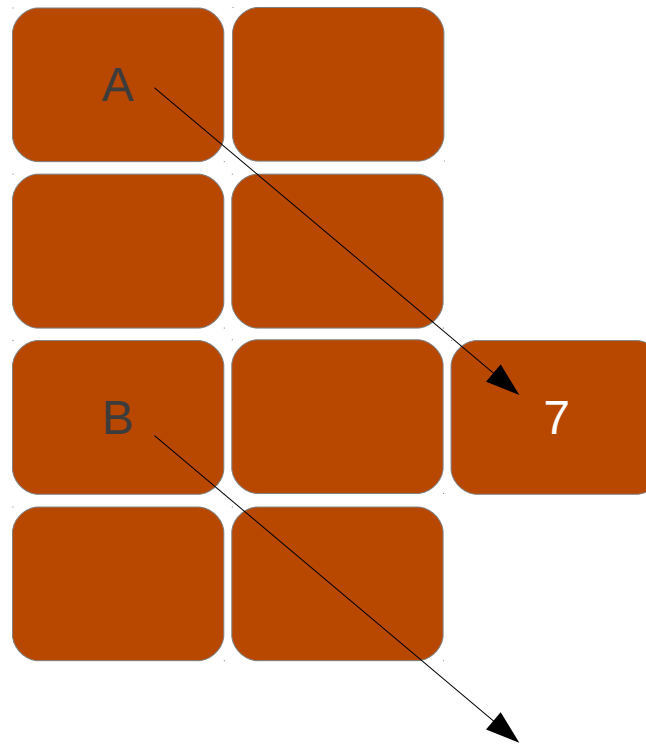
Phase 1. 対応関係

- 枝刈りする場合



Phase 1. 対応関係

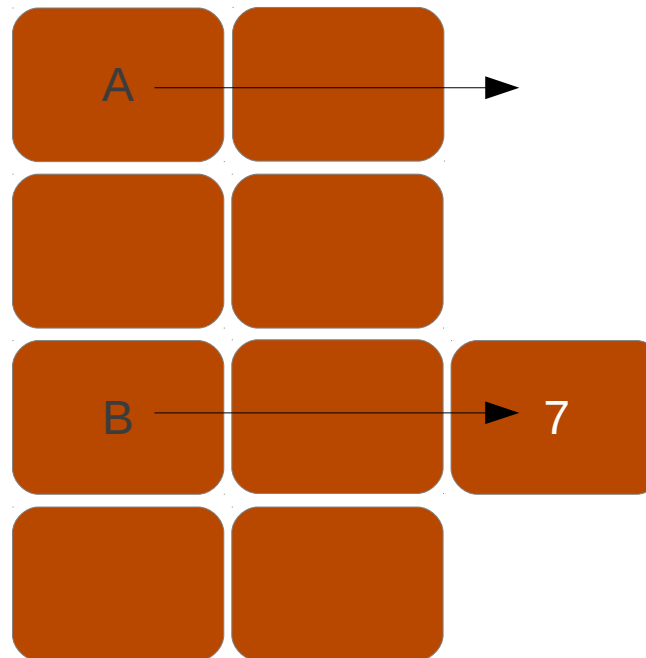
- 枝刈りする場合



pair[7] がない

Phase 1. 対応関係

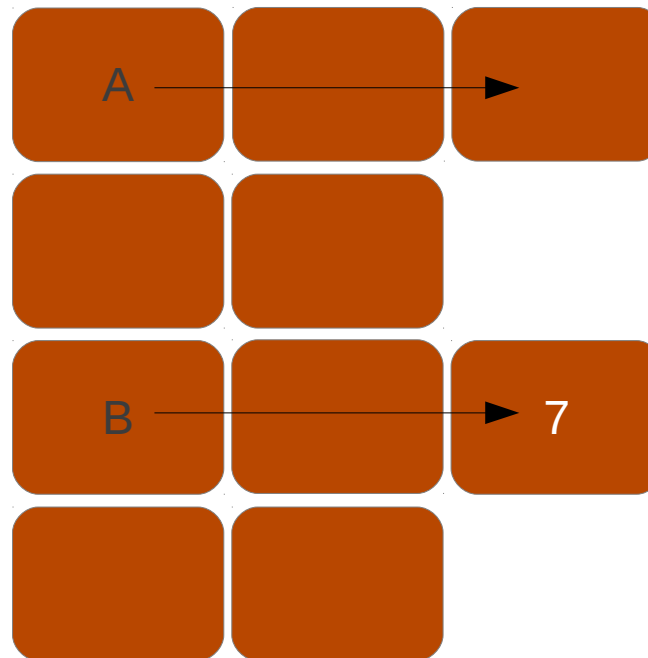
- 枝刈りする場合



rpair[7] もない

Phase 1. 対応関係

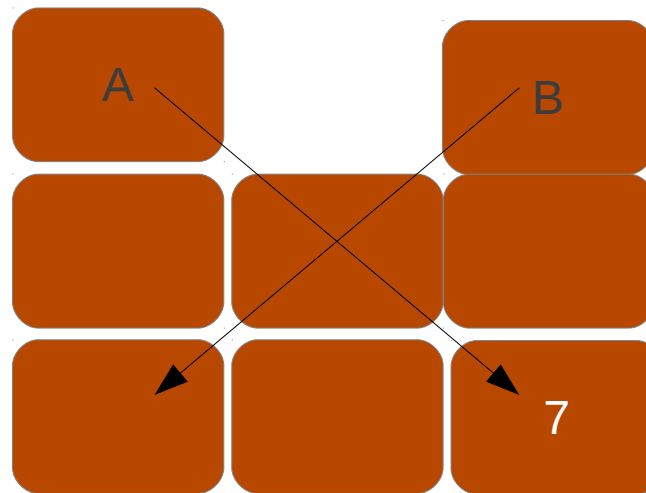
- cf. 枝刈りしない場合



rpair[7] がある

Phase 1. 対応関係

- cf. 枝刈りしない場合



pair[7] がある

Phase1 は通るが正解は NO

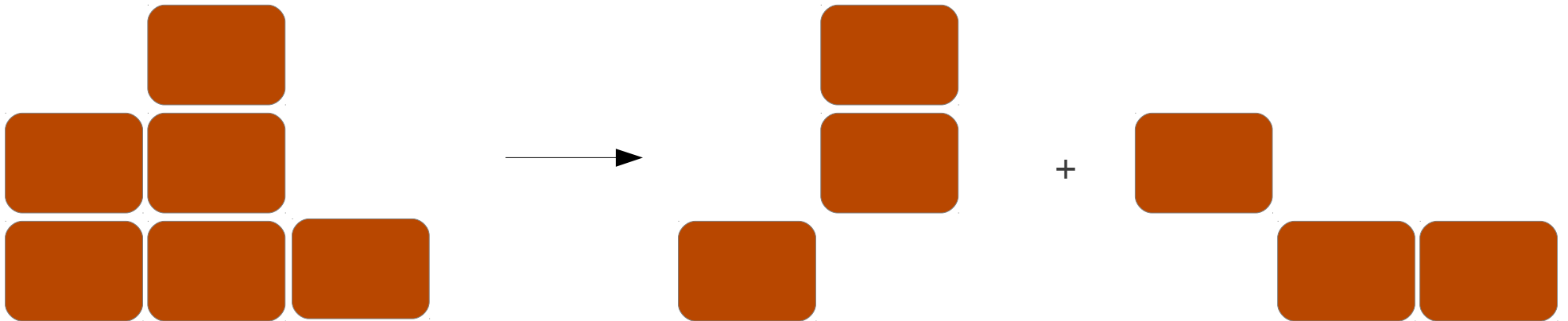
Phase 1. の計算量

- B の基準点 N 通り
- B の座標系 8 通り
- 全てのピースについて対応を調べる
 - N 通り

$$O(N^2)$$

Phase 2. 隣接関係

- 対応関係さえあれば Phase1 を通る
- こういうのを排除



Phase 2. 隣接関係

- 深さ優先探索
 - スライドの最初のやつを行う
- ビット集合表現の有効利用

Phase 2. の計算量

- 隣接してたら追加してみる
 - $O(N * N/2)$ だが ...
- メモ化探索が使える
 - ある形のとときに無理と判明したら保存
次にその形になったら探索を打ち切る
- Phase 1 で大分枝刈りできている

ビット集合表現

- ビットをフラグとして用いる
- i 番目を取り出す
 - $(S \gg i) \& 1$
- 和集合
 - $S | T$
- 共通集合
 - $S \& T$

1 =

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

S	0	1	0	0	1	0	0	1
T	1	1	0	1	0	0	1	0