

突然ですが、

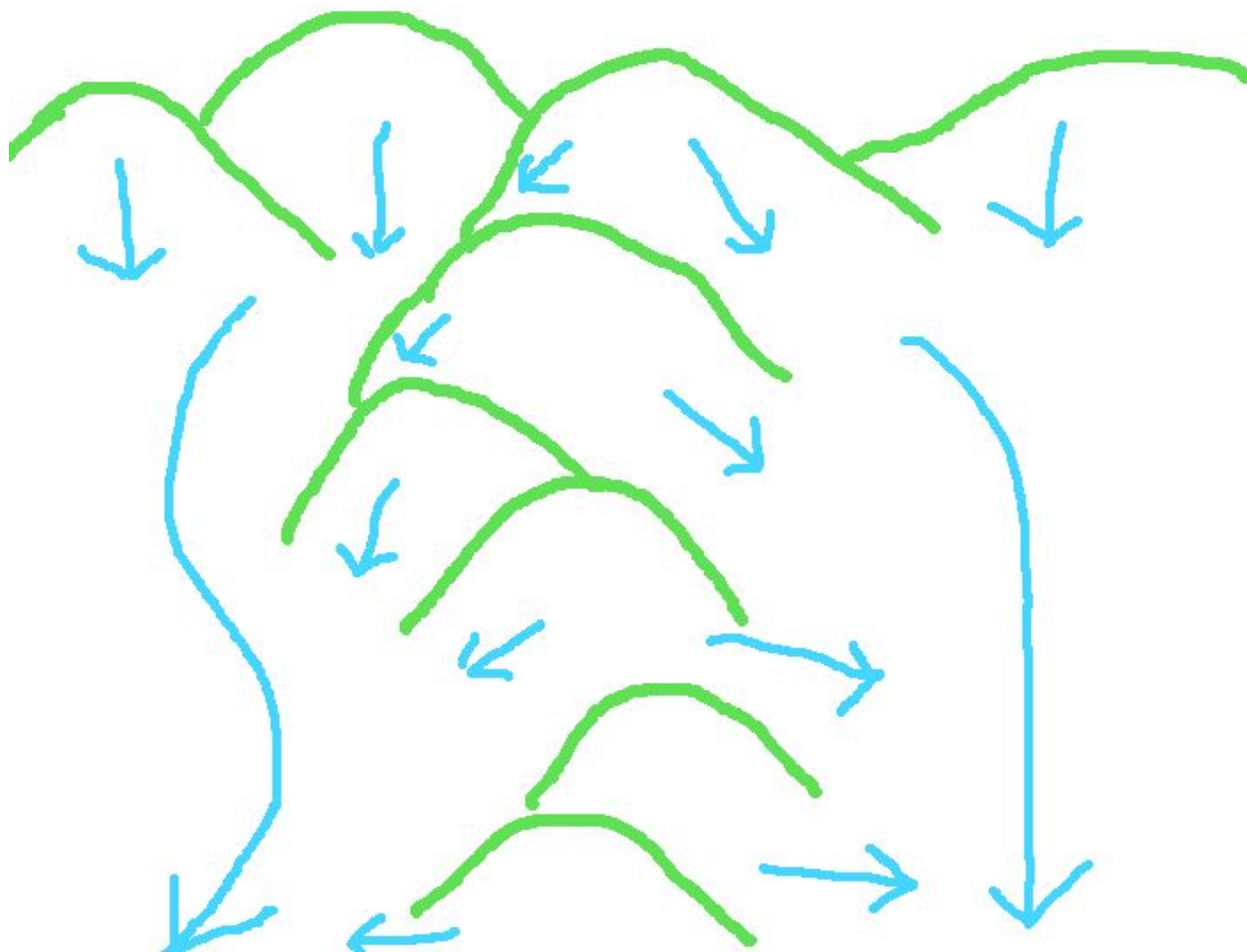
分水嶺って、よくね？

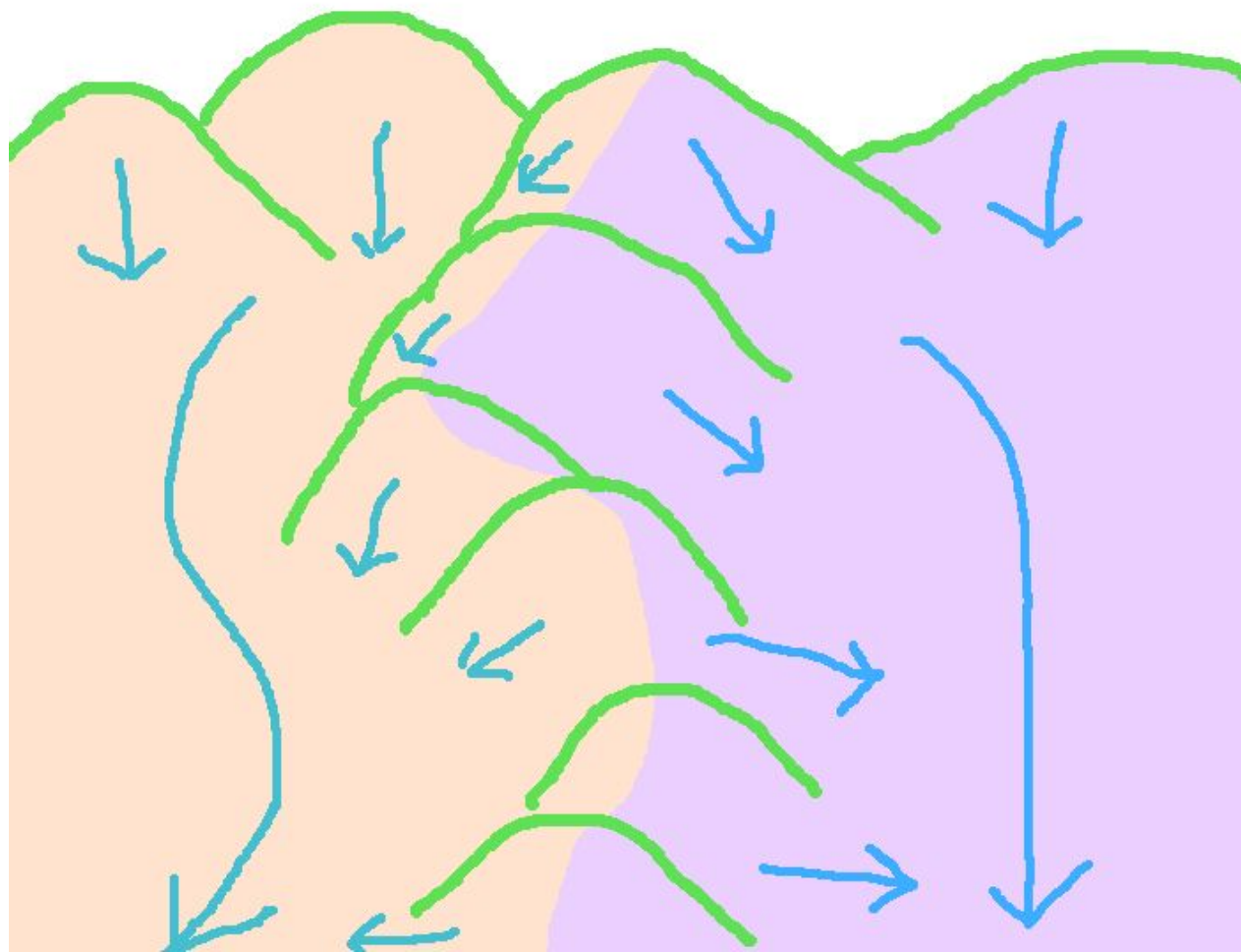
分水嶺 = 水系界の尾根

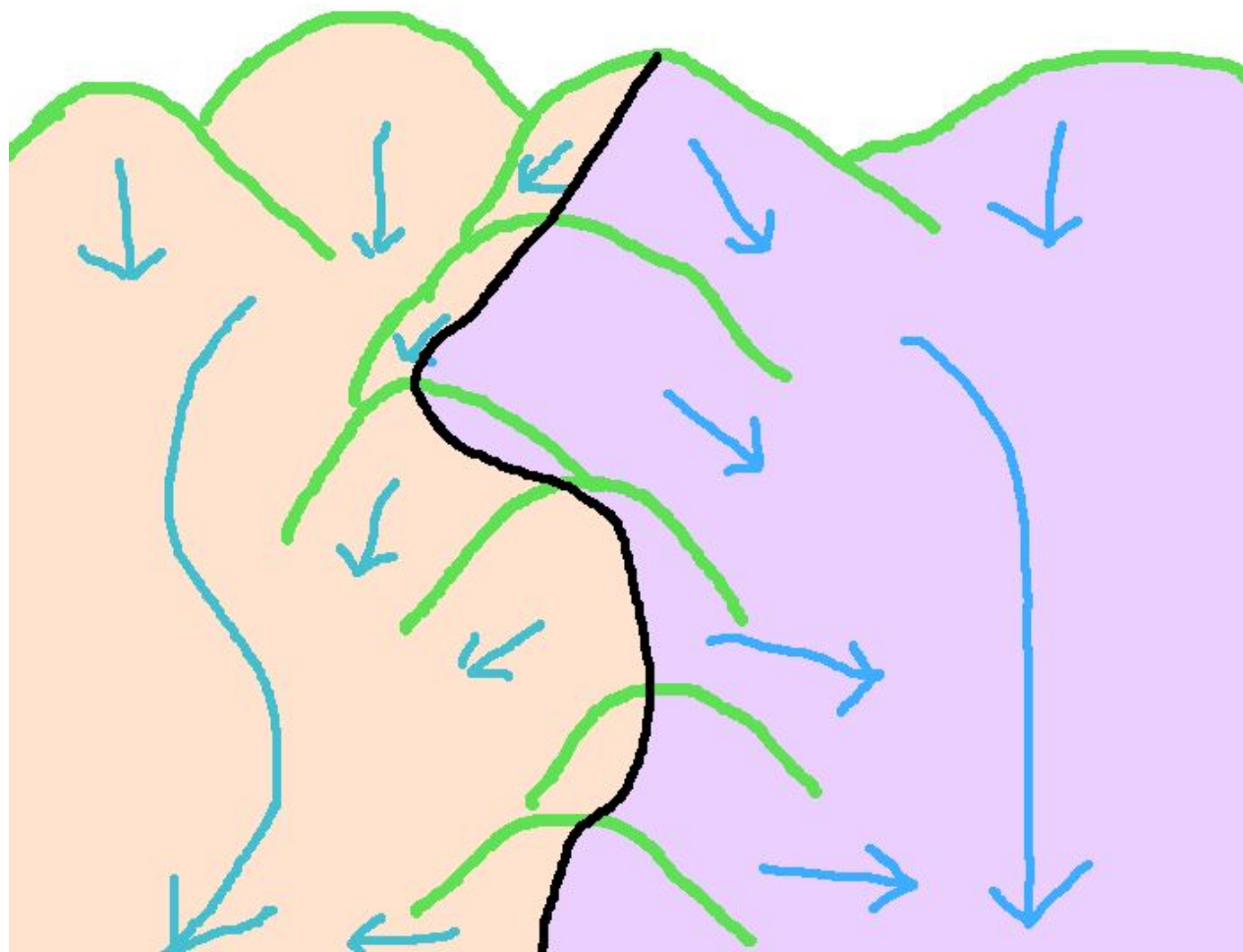
水系 #とは











例題1



例題1 (水系分類はいいかんじに修正されています)



例題2



例題2 (水系分類はいいかんじに修正されています)



例題3



例題3 (水系分類は (r y



Q. どうやって探す？

Q. どうやって探す？

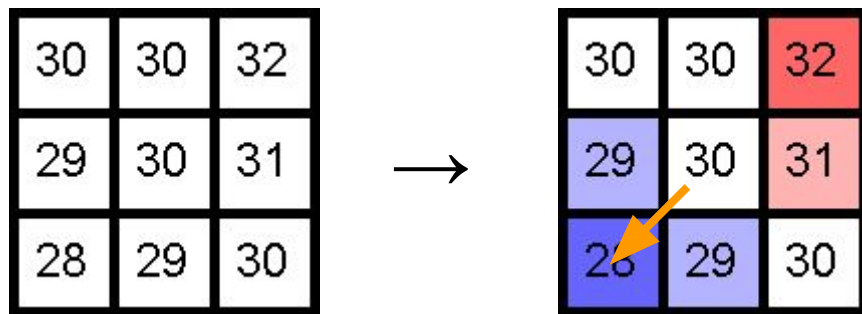
A. 愚直にやる。

Step1. データを手に入れてくる

- 国土地理院のAPIをたたけばOK
- <https://maps.gsi.go.jp/development/ichiran.html>
- 地図タイル(.png)と標高データ(.csv)が手に入る
- 今回はDEM10Bを使用
- 地図1ピクセル = 標高データ1マス (やったね)
- パースと結合は、やるだけ。

Step2. 水の行き先を決める

- 水は低い方に流れる（それはそう）
- 各マスについて、近隣の一番低いマスを探す



Step2. 水の行き先を決める

30	30	31
30	30	31
30	30	30

- もし低いマスが見つからなかったら？：
- 探索範囲を上下左右1マスずつ広げる (最大20)

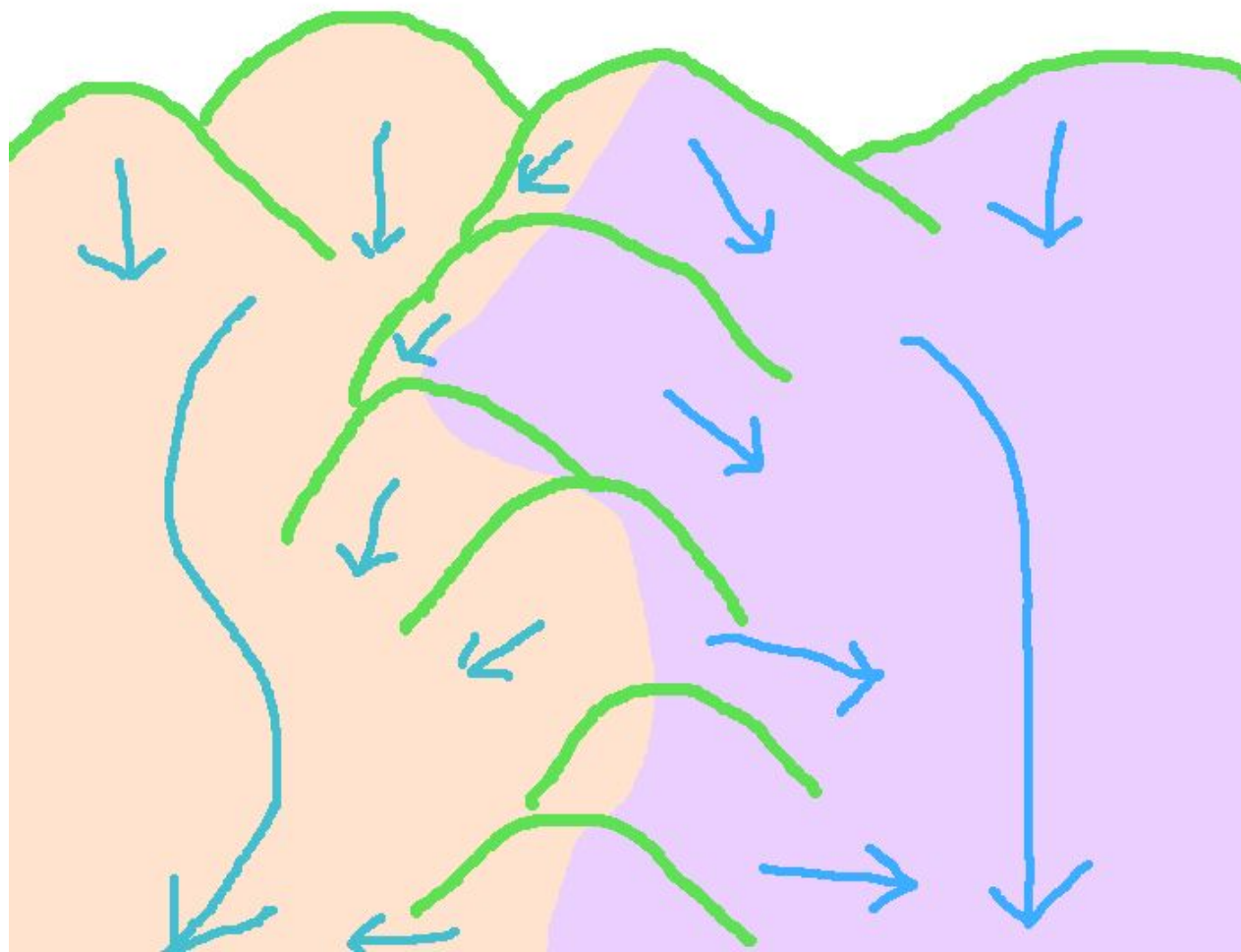
30	30	30	31	32
30	30	30	31	32
30	30	30	31	31
30	30	30	30	31
29	30	30	30	30



30	30	30	31	32
30	30	30	31	32
30	30	30	31	31
30	30	30	30	31
29	30	30	30	30

Step3. 水系を分類する

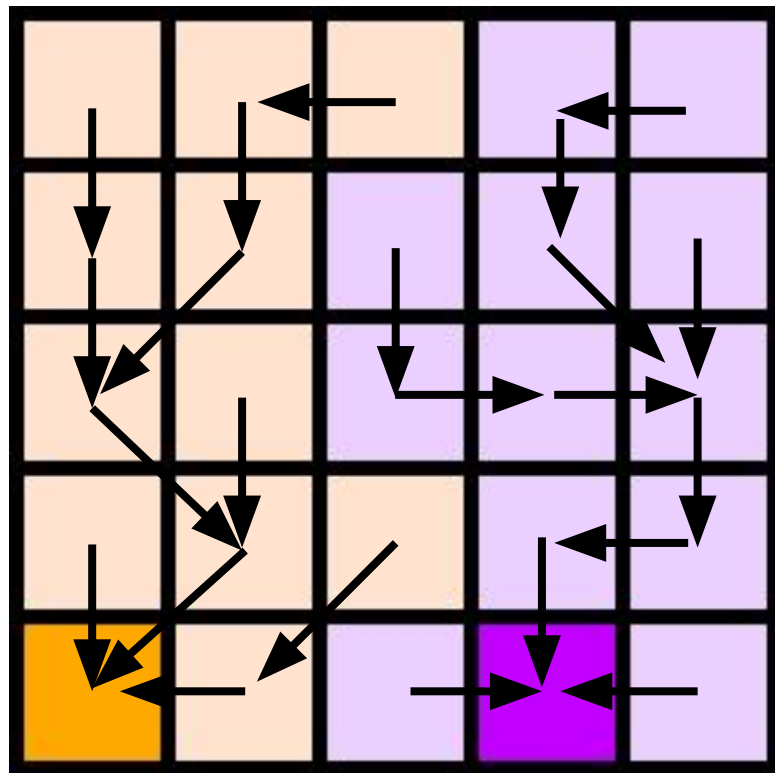
- すべてのマスについて行き先を辿る
- 最後にたどり着いたマスで分類



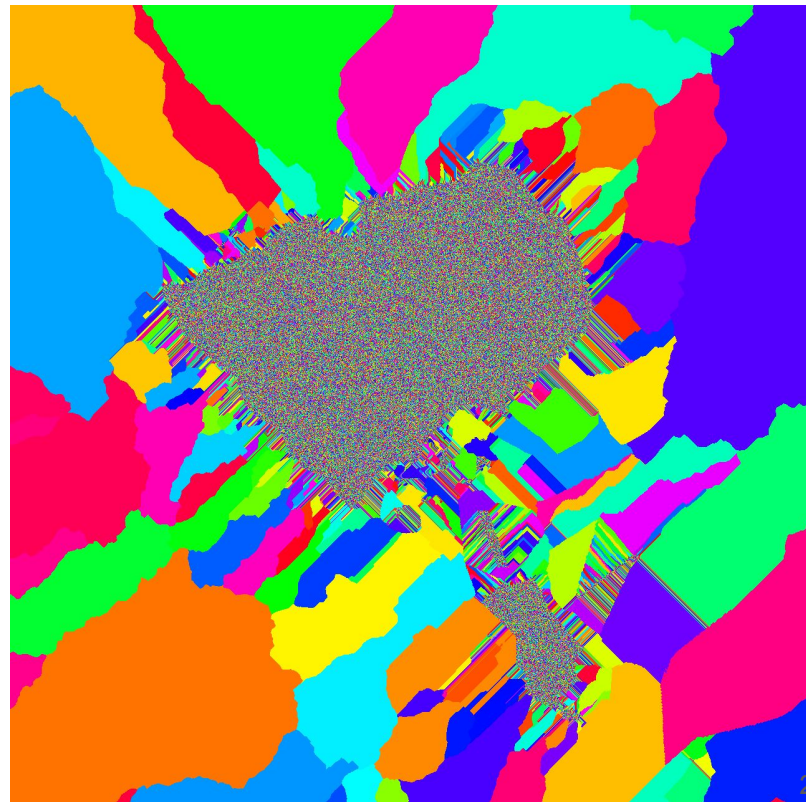
Step3. 水系を分類する

- すべてのマスについて行き先を辿る
- 最後にたどり着いたマスで分類
- ループは存在しない（たぶん）
- 愚直にやると $O(n^{1.5})$ （たぶん）で遅い
- DPすれば線形時間

Step3. 水系を分類する



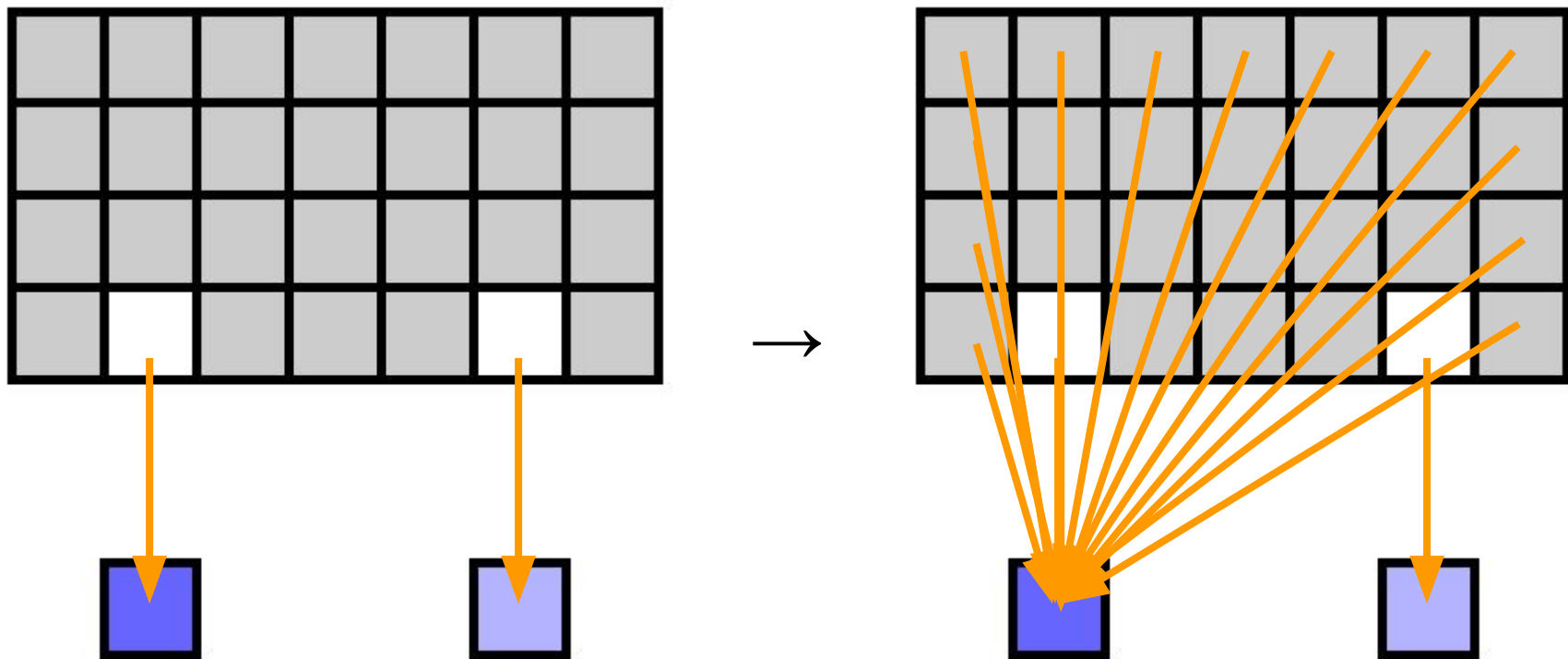
うごかしてみた



Step2'. 行き止まりを修正する

- 短径20マス以上の水域に行き止まりが大量発生した
- そうだ 併合、しよう。
- 行き止まりマス塊の行き先を次のように設定：
 - 塊に隣接するマスの行き先のうち最も低いマス

Step2'. 行き止まりを修正する

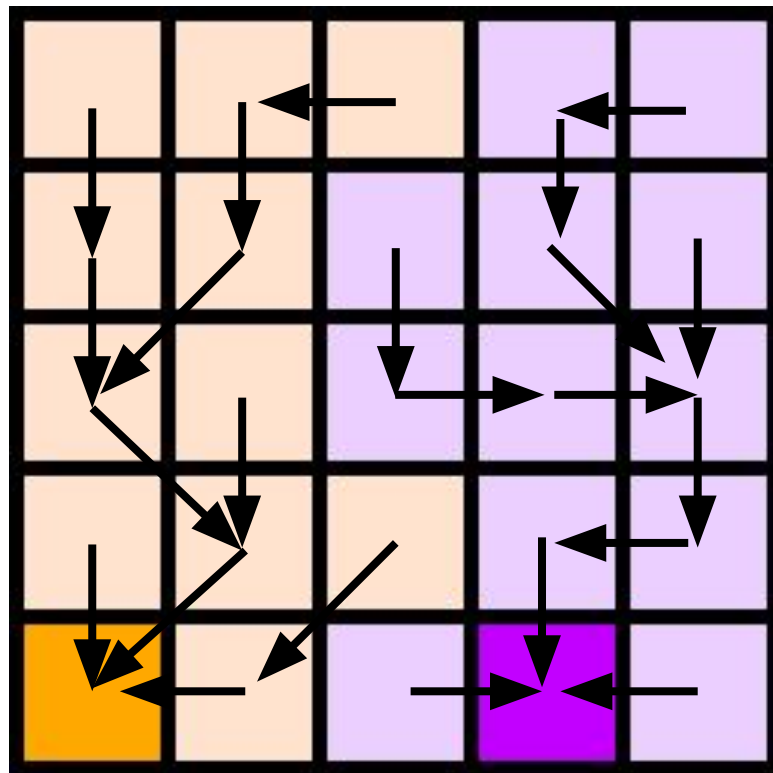


修正の効果



Step3. 水系を分類する

この情報があったら →
もっとなにかできそう

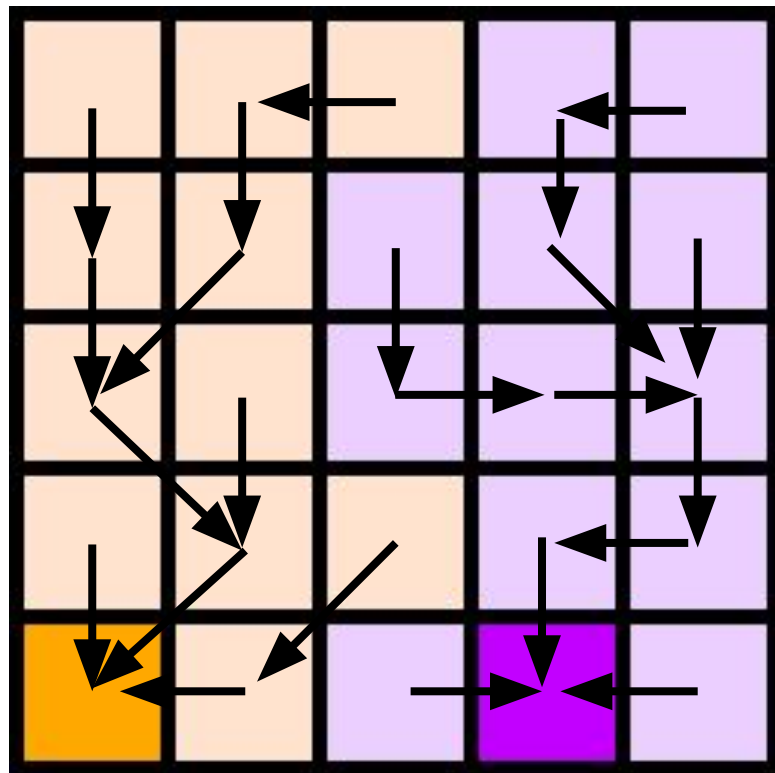




もし雨が均等に降ったら

Step3. 水系を分類する

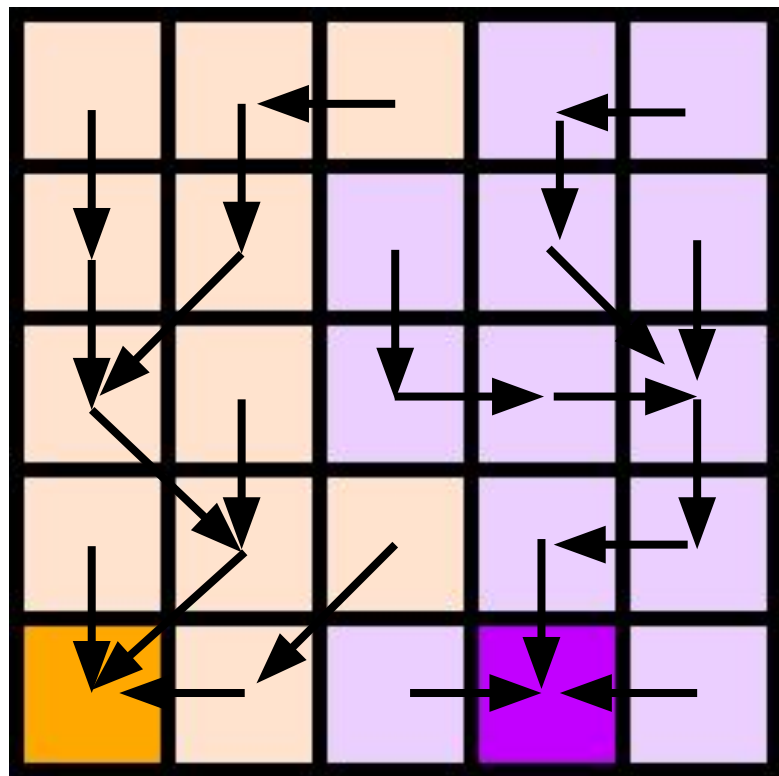
あるマスの水量
= それより上流のマスの数



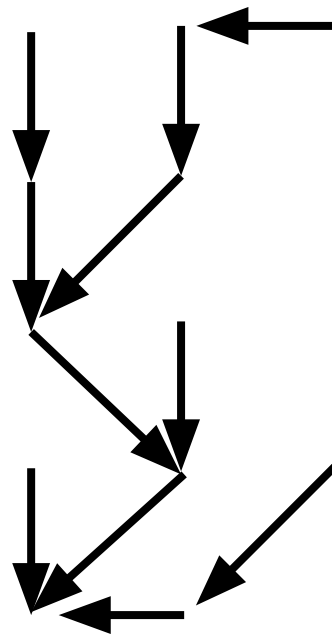
Step3'. 水量をカウントする

- すべてのマスについて行き先を辿る
- 辿りながら上流のマス数を合計していく
- ↑愚直にやれば余裕（だが遅そう）
- DPするには工夫が必要
- そうだ トポロジカルソート、しよう。

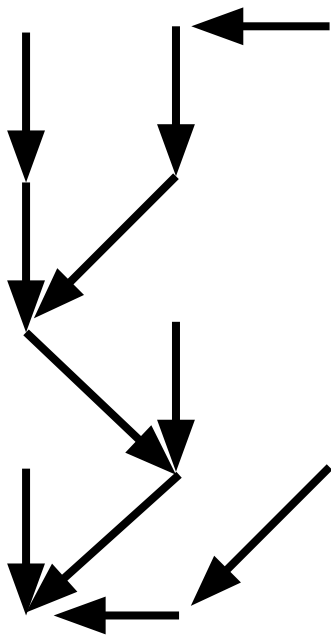
Step3'. 水量をカウントする



説明のために
1水系だけ取り出す

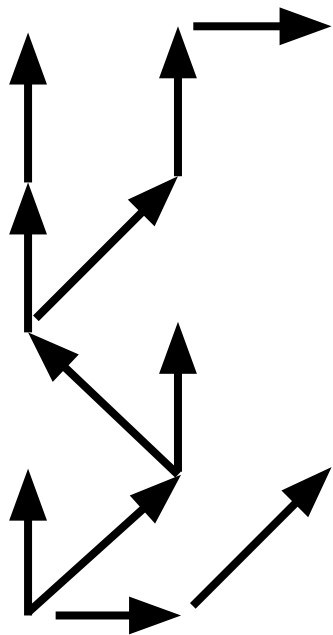


Step3'. 水量をカウントする



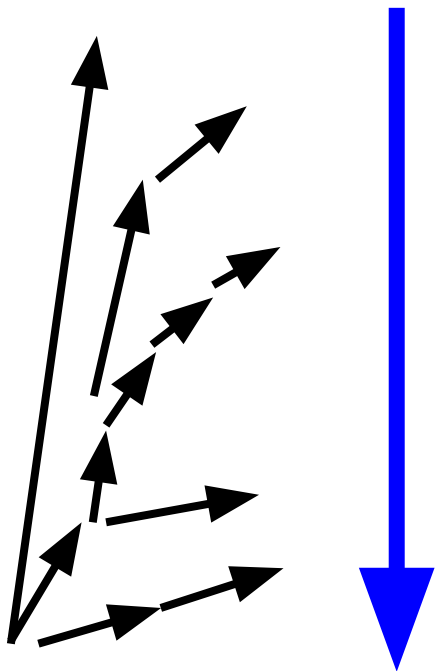
- 親ノードが子ノードを知らない
- 扱いづらいため、親に子を教える

Step3'. 水量をカウントする



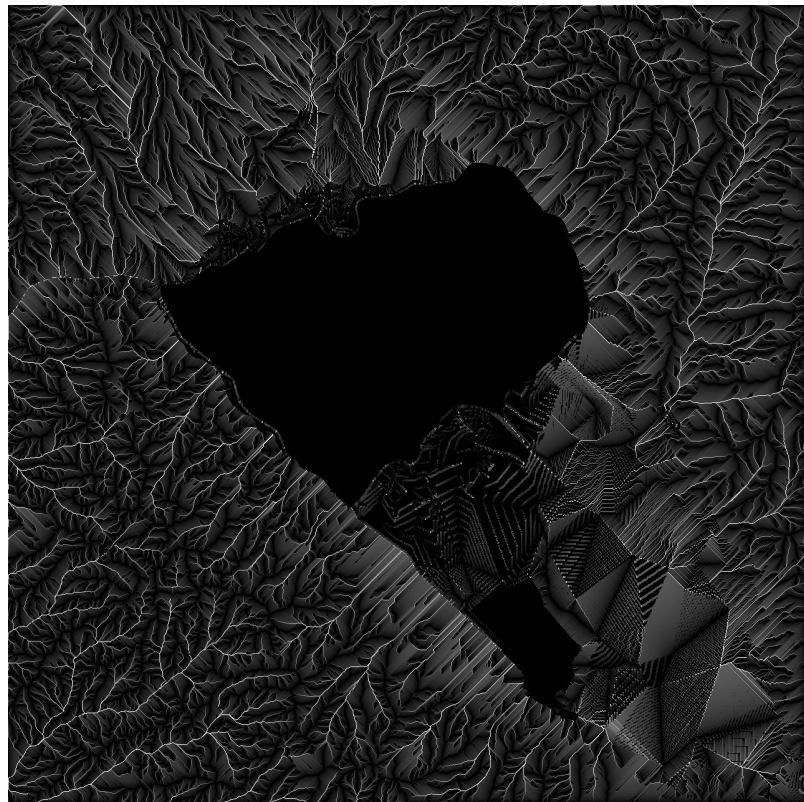
- 親ノードが子ノードを知らない
- 扱いづらいので、親に子を教える
- 子から親へ走査すればDPできる
- 根からbfs/dfsして並べ替え

Step3'. 水量をカウントする



- 親ノードが子ノードを知らない
- 扱いづらいので、親に子を教える
- 子から親へ走査すればDPできる
- 根からbfs/dfsして並べ替え
- 探索時と逆順に進みながらカウント
- 愚直にやるより速い！

計算した水量



水量をもとに透過度を設定して合成

