A - 問題 & 解法

問題

- aにいくつかの数x(x≧0)足して、bの倍数にしたい. xの最小値を出力.

- 解法
  - a%b==0のとき、0をそのまま出力
  - a%b≠0のとき、b-a%bを出力
- 別解
  - bが小さいのでforループでaがbの倍数になるまでインクリメントしていってもよい

B - 問題 & 解法

問題

- n個の商品の価値と 整数Xが与えられる．Xのk番目のビットがたっているとき、k番目の商品を選ぶとする．選ぶ価値の合計を出力しなさい．

- 解法
  - K番目のビットがたっているかどうかの判定は色々あるが、ビット演算を用いるとよい．たとえばC言語だと、if(X >> k & 1) { 立ってるときの処理 } とかでかける.
  - それを用いて，すべてのk(0≦k≦n-1)についてループで価値を足し合わせる

C - 問題 & 解法 (1/3)

問題

- 区間がたくさん与えられる.
  もっとも区間に被覆されてる頂点のその被覆数を出力しなさい.

考察

- すべての点について確かめるのは時間がかかりそう

- 部分点解法
  - いずれかの区間の端点を調べるだけで十分(ほかは無駄)
  - すべての端点(n×2通り)について、含んでいるものがいくつあるかループで調べる
  - 端点の数は2n個，区間の数はn個　∴時間計算量O(n^2)
- 満点解法
  - 値の上限をk=1,000,000としてO(n+k) →次ページ

C - 問題 & 解法 (2/3)

満点解法

- 一般にいもす法と呼ばれる累積和の応用テクを用いる
- 大きさ1000001個の点に対応する配列Sを用意する.
  前処理として、ある区間[a,b](aとbを含む閉区間)を追加するとき
  S[a]++; S[b+1]--;としておく.
- 全ての区間を前処理したあと、

  for(i=1;i<=1000000;i++) s[i] += s[i-1]; // 0-indexed

  のように累積和をとる.
- 配列Sの各要素にはそれぞれの点の被覆数が格納されている
- 時間計算量O(n+k)
- 今回は必要ないが、座標圧縮(出現する点だけを残す手法)をすれば
  O(n)になり、区間の端点が取る値の大きさによらない

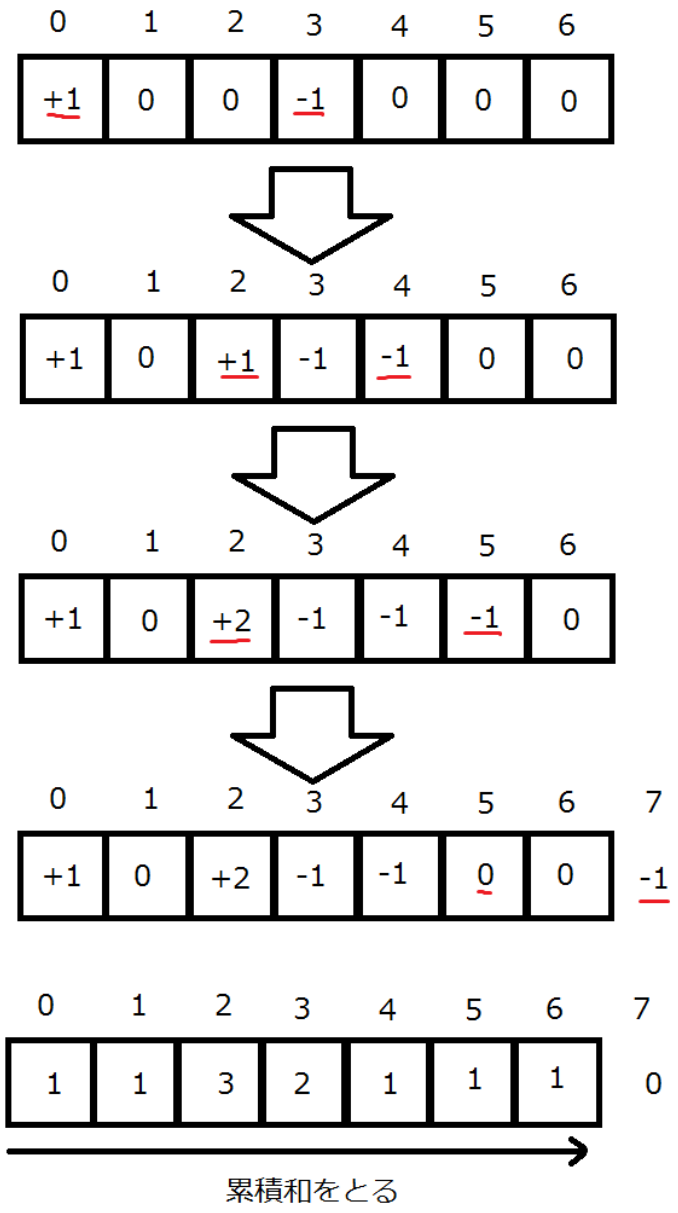
- →次のページに動作例

C - 問題 & 解法 (3/3)

動作例

- [0,2],[2,3],[2,4],[5,6] (sample1)

- I. s[0]に+1,s[3]に-1
- II. s[2]に+1,s[4]に-1
- III. s[2]に+1,s[5]に-1
- IV. s[5]に+1,s[7]に-1
- V.累積和をとる
- 一番被覆されてるのは
  2番目で被覆数3



D - 問題 & 解法 (1/4)

問題

- 木に1つの辺を追加するとき，できるループの大きさを出力．

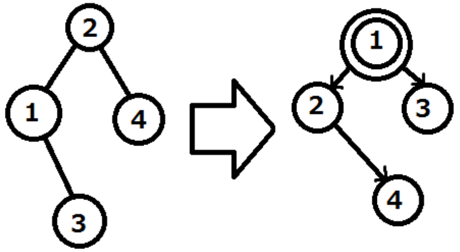
部分点解法

- 与えられる追加辺(a,b)について，aを始点，bを終点とし探索(幅優先探索でも深さ優先探索でも)して得た経路長に+1したものが答え(探索時，追加辺は考慮しない)

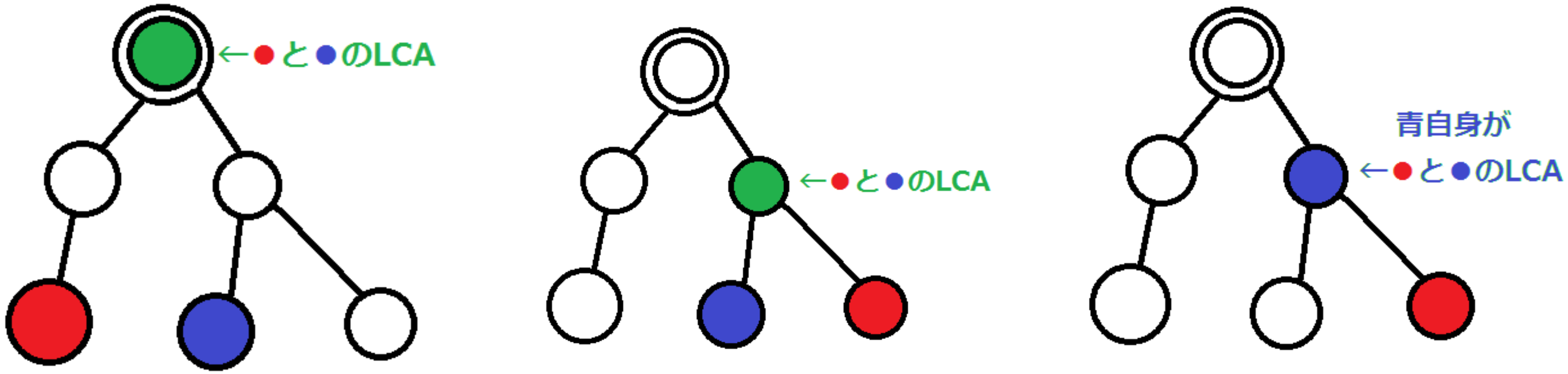
- 満点解法
  - LCAを用いて，a↔b間の最短距離を求めた上で+1したものを出力すればよい
  - →詳しくは次ページ

D - 問題 & 解法 (2/4)

- 満点解法
  - LCAを用いる
  - 与えられるグラフを，適当な頂点(どこもいい)を根とした根付き木として扱う.
  - この根付き木について，全ての頂点の深さとその親を配列に格納しておく(ただし根の深さは0で親はいない) .
  - LCA(最小共通祖先)と呼ばれるものを高速に計算する.
  - LCAとは，2つの頂点の共通の祖先(親を巡ってたどり着ける頂点)で最も近いもの



頂点1を根とする場合



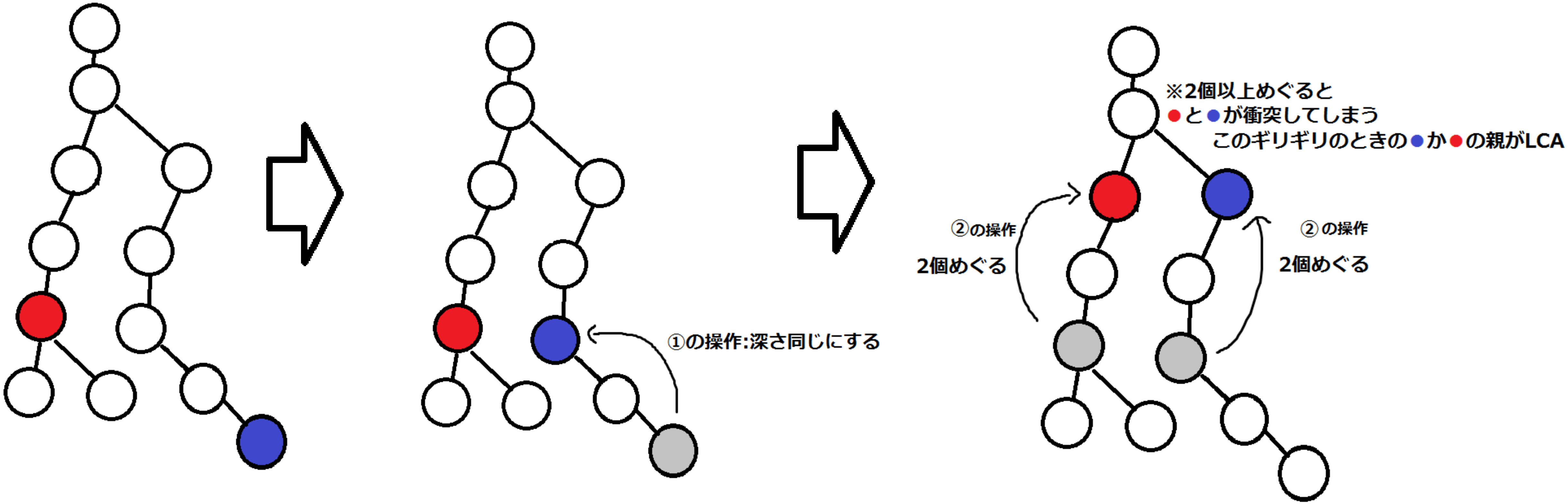
D - 問題 & 解法 (3/4)

- それぞれの頂点について，2^k個前の親を予め計算して保持しておく(テーブルをつくる)　いわゆるダブリング
- ある頂点xの2^k(k>0)個前の親 =
  {xの2^(k-1)個前の親}の2^(k-1) 個前の親
  (存在しない場合は場合分け)
- ということを利用するとこれらはkが小さい場合から全頂点について逐次計算していけば求まる
- そして，LCAを求める際にそのテーブルを用いる.
- 具体的には頂点a,bのLCAを求めるとき，
  ①aとbが同じ深さになるまで片方を登らせた上で，
  ② 2者が衝突しないギリギリの高さまで2者を登らせる
  を求める(①と②の操作はどちらもテーブルを利用してlog nで可能．)

D - 問題 & 解法 (3/4)

視覚的な例

①aとbが同じ深さになるまで片方を登らせた上で,
② 2者が衝突しないギリギリの高さまで2者を登らせる



D - 問題 & 解法 (4/4)

①aとbが同じ深さになるまで片方を登らせた上で,
② 2者が衝突しないギリギリの高さまで2者を登らせる

- これらをO(log n)で行う
- 詳しくは正解者のコードを見ると良さそう(すみませんまだかけてません...).

- 出力すべきは(aの深さ) + (bの深さ) – (aとbのLCAの深さ) + 1