A問題 解説
「コード川柳」

CODE FESTIVAL 2015 本戦

概要と解説

- 文字列が 3 つ与えられます
- 長さが 5 、 7 、 5 になっているかどうかを判定してください

- まず、文字列を入力します
- 文字列の長さを調べます
- もし間違った長さのものがあれば NO と出力します
- そうでなければ YES と出力します

B問題 解説
「ダイスゲーム」

CODE FESTIVAL 2015 本戦

問題概要

- ６面ダイスをN個振ったとき、和として最も出やすい値を求めよ
- 複数存在する場合はそのうち最も小さい値を求めよ

- $1 \leqq N \leqq 256$

解法

- 確率の分布を考えると、和として出うる値のうちの中央値が最も出やすい
- 7*N/2（小数点以下切捨て）が答え

- ただし、N=1のときは1~6が出る確率が同じなので、答えは1となる

C問題 解説
「寿司タワー」

CODE FESTIVAL 2015 本戦

問題概要

- 寿司をN個を、ネタとシャリ（合計2N個）がある順番で並ぶように積む
- 寿司の積み方は
  - そのまま積む：シャリ、ネタの順に積まれる
  - ひっくり返して積む：ネタ、シャリの順に積まれる
  - 分解して積む：ネタとシャリを別々に積む
- 分解しなければならない寿司の個数の最小値は？

- 制約
  - $1 \leqq N \leqq 256$

解法

- できるだけ分解しない寿司を多くしたい
- 分解せずに積める寿司の個数の最大値を求めたい
- 逆に、ネタとシャリを合体させて寿司を作っていくと考える
- 下から順に見ていき、合体させて寿司が作れる部分を見つけたら寿司を作っていく

シャリ
シャリ
ネタ
ネタ
シャリ
ネタ

解法

- できるだけ分解しない寿司を多くしたい
- 分解せずに積める寿司の個数の最大値を求めたい
- 逆に、ネタとシャリを合体させて寿司を作っていくと考える
- 下から順に見ていき、合体させて寿司が作れる部分を見つけたら寿司を作っていく

シャリ
シャリ
ネタ
ネタ
シャリ
ネタ

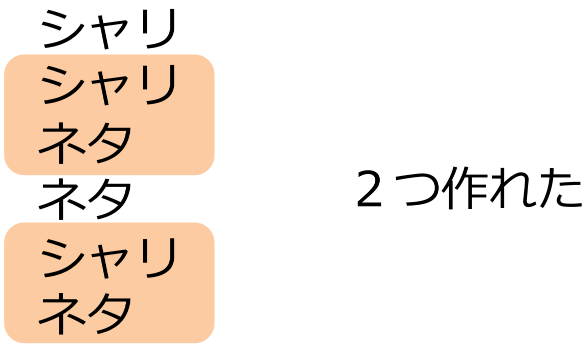
解法

- できるだけ分解しない寿司を多くしたい
- 分解せずに積める寿司の個数の最大値を求めたい
- 逆に、ネタとシャリを合体させて寿司を作っていくと考える
- 下から順に見ていき、合体させて寿司が作れる部分を見つけたら寿司を作っていく

シャリ
シャリ
ネタ
ネタ
シャリ
ネタ

解法

- できるだけ分解しない寿司を多くしたい
- 分解せずに積める寿司の個数の最大値を求めたい
- 逆に、ネタとシャリを合体させて寿司を作っていくと考える
- 下から順に見ていき、合体させて寿司が作れる部分を見つけたら寿司を作っていく



解法

- 「N - 作ることができた寿司の個数」が答え
- バグを出さないように注意して実装しましょう

D問題 解説
「足ゲームII」

CODE FESTIVAL 2015 本戦

問題概要

- 区間がN個与えられる
- １つだけ区間を取り除くことができる
- 同じ場所に重なっている区間の個数の最大値の最小値を求めよ

- 制約
  - $2 \leqq N \leqq 10^5$
  - $1 \leqq$ 区間の端点の座標 $\leqq 10^5$

解法

- １つも区間を取り除かないときの「同じ場所に重なっている区間の個数の最大値」を m とします
- 区間を１つ取り除いても「同じ場所に重なっている区間の個数の最大値」は高々1しか減らないため、答えは m または m-1 になる
- 区間を１つ取り除いて「同じ場所に重なっている区間の個数の最大値」を減らすことができるかどうかを判定できれば良い

解法

- 「同じ場所に重なっている区間の個数の最大値」を減らすことができるような区間とはどんな区間？
  - m 個の区間が重なっている部分を全て覆っているような区間

解法

- m 個の区間が重なっている部分を全て覆っているような区間の見つけ方
  - まず、各部分をいくつの区間が覆っているかを求める
    - imos法（累積和）などを用いて計算できる
  - 次に m 個の区間が重なっている部分を列挙する
  - そのうち、最も左にあるものと右にあるものを両方覆うことができる区間を探す。

- 計算量は O(N) となる

E問題 解説
「ショートコーディング」

CODE FESTIVAL 2015 本戦

問題概要

- 単項演算子「-」「!」のみからなる式が与えられる
  - 「-」：正負を反転する
  - 「!」：0を1に、それ以外を0にする
- -256~256 の入力についての出力が全て同じになるような式のうち、最も短いものを求めよ

- 1 ≦ 式の長さ ≦ 256

解法

- 考察
  - 「!」の演算結果には入力の正負は関係ない
    - →「!」の後ろにある「-」はあってもなくても変わらない
    - そういうものを消すと「-----!!!!」のように、「-」がx個並んだ後に「!」がy個並んだ式になる

解法

- 考察
  - 「--」は消して良い
    - 「-」がx個並んだ式は「-」がx%2個ならんだ式に書き換えてよい

解法

- 考察
  - 「!!」は基本的に消して良い
    - ただし、「!!」と「」は別の結果となるため、残りが２個の場合は消してはいけない
    - つまり「!」がy個ならんだ式は、
      - y=0：「」
      - y>0、yが奇数：「!」
      - y>0、yが偶数：「!!」

解法

- 解法
  - まず、「!」の後ろにある「-」を消す
  - 「--」を消していく
  - 「!」の個数が2以下になるまで「!!」を消していく

- ちなみに、答えは一意に定まります

F問題 解説
「歩くピアニスト」

CODE FESTIVAL 2015 本戦

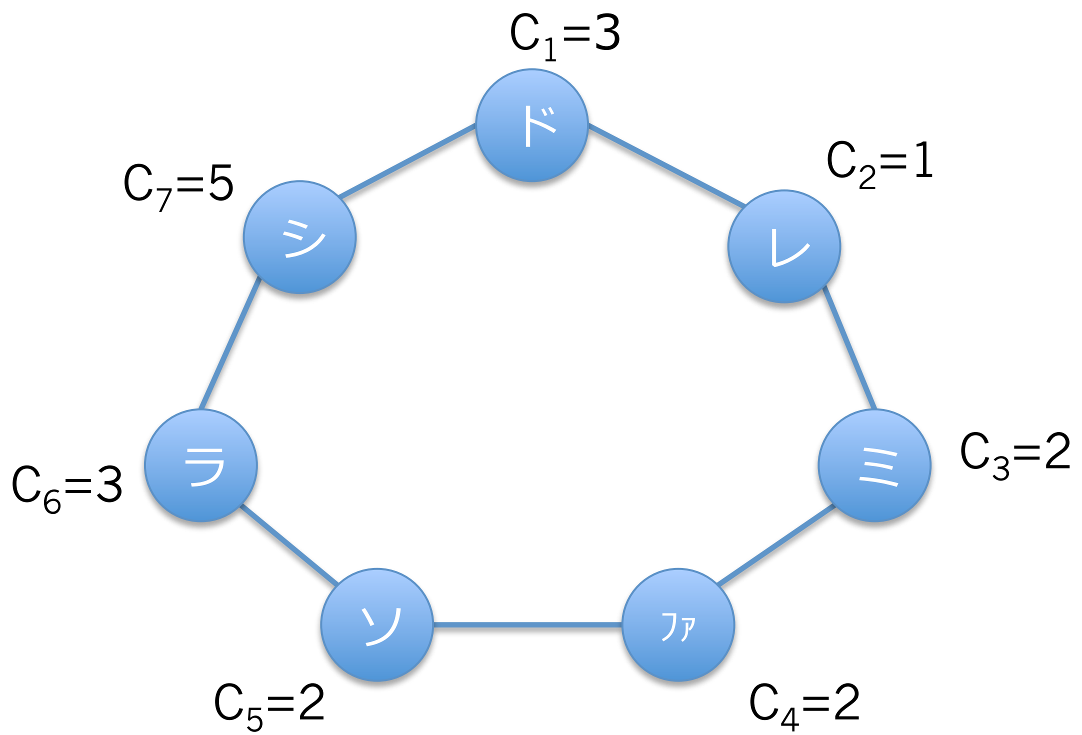
問題概要

- 巨大なピアノがある
- 以下のような演奏ができるかどうかを判定せよ
  - ド～シをそれぞれちょうどC_1~C_7回鳴らす
  - ある音の直後には隣り合った音しか鳴らすことができない
    - 例えばド→レ、ファ→ミ、ド→シなどはOK
  - ドから始めて、ドで終わる

- 制約
  - $0 \leqq C_1 \sim C_7 \leqq 10^{10}$
  - C_1~C_7の和は 1 以上

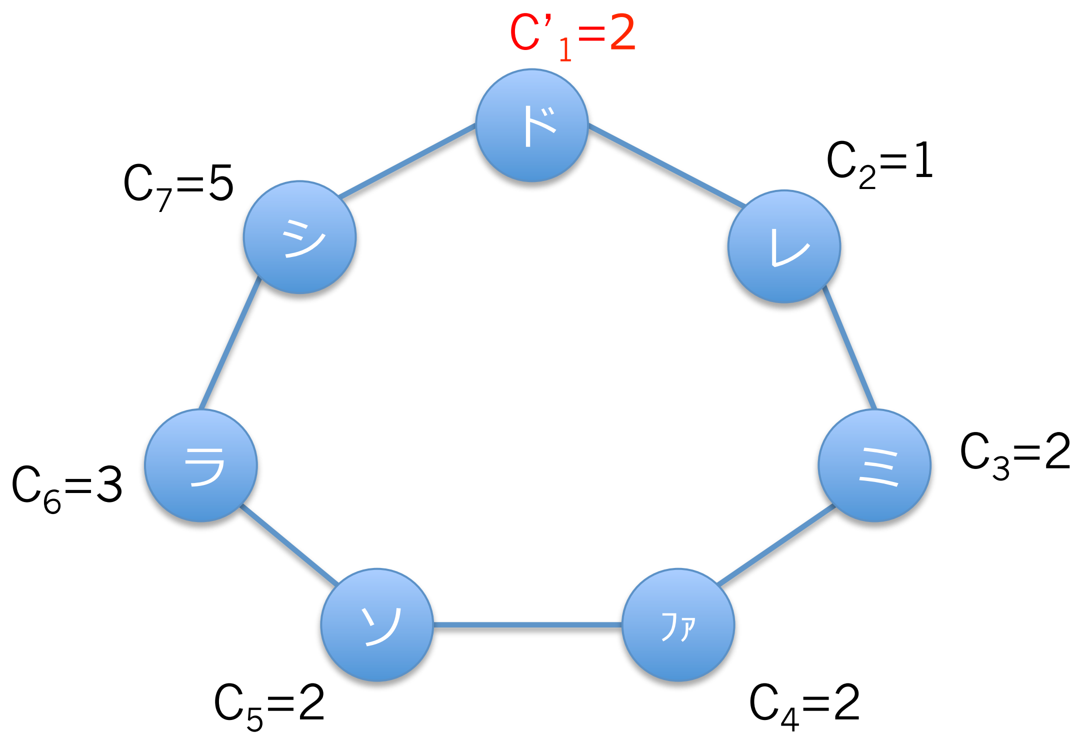
解法

- グラフ問題に帰着する
  - 頂点は音階、辺は隣り合った音を結ぶ
  - ドからはじめてドで終わるようなパスで、各頂点を通る回数がそれぞれちょうどC_1~C_7回になっているようなものがあるか



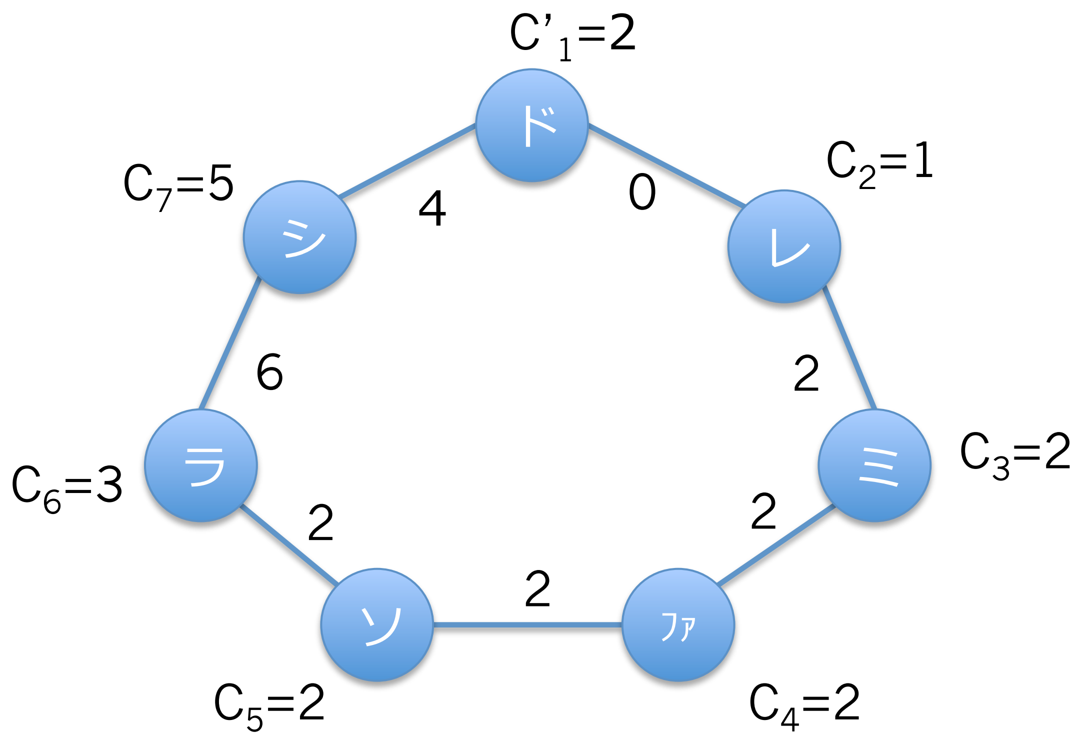
解法

- 計算しやすいように、少し言い換える
  - ドからはじめてドで終わるようなパスで、各頂点を通る回数がそれぞれちょうどC_1~C_7回になっているようなものがあるか
  - →ドを含むようなサイクルで、各頂点を通る回数がそれぞれちょうどC_1-1,C_2~C_7回になっているようなものがあるか



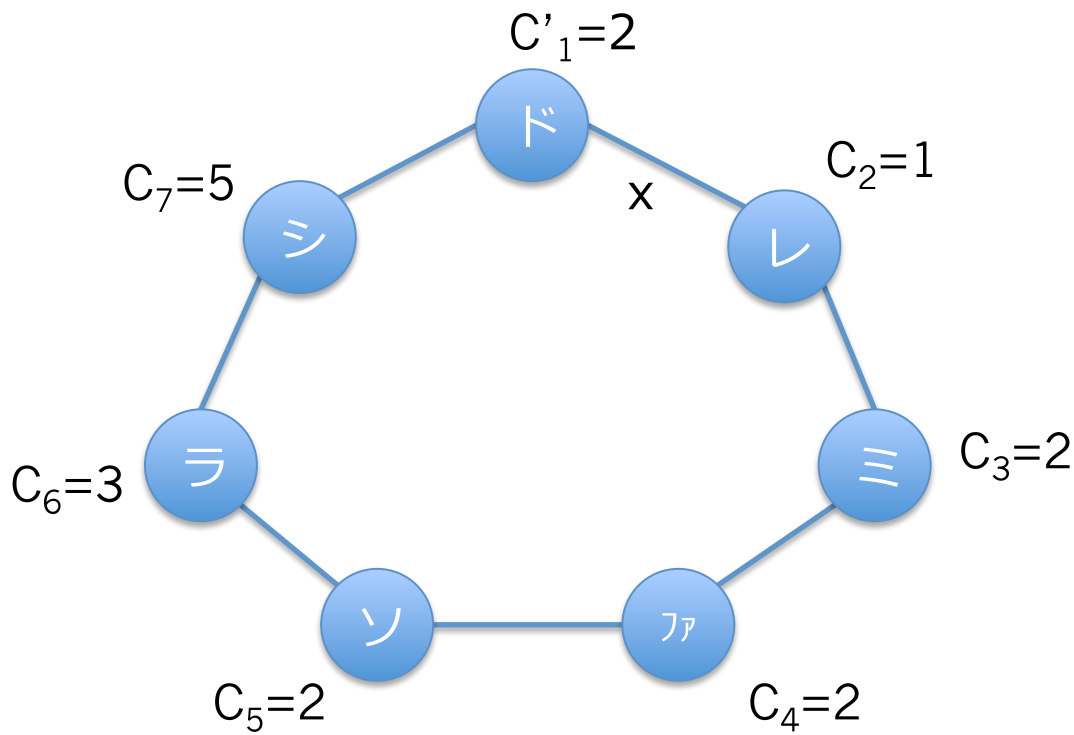
解法

- 辺を通る回数に注目する
  - 辺を通る回数はどういう条件を満たせばいいか
    - 各頂点を踏む回数*2 = 隣接した 2 つの辺の通る回数の和
      - 各頂点は 1 回の訪問につき入るときと出るときで 2 回辺を使うため



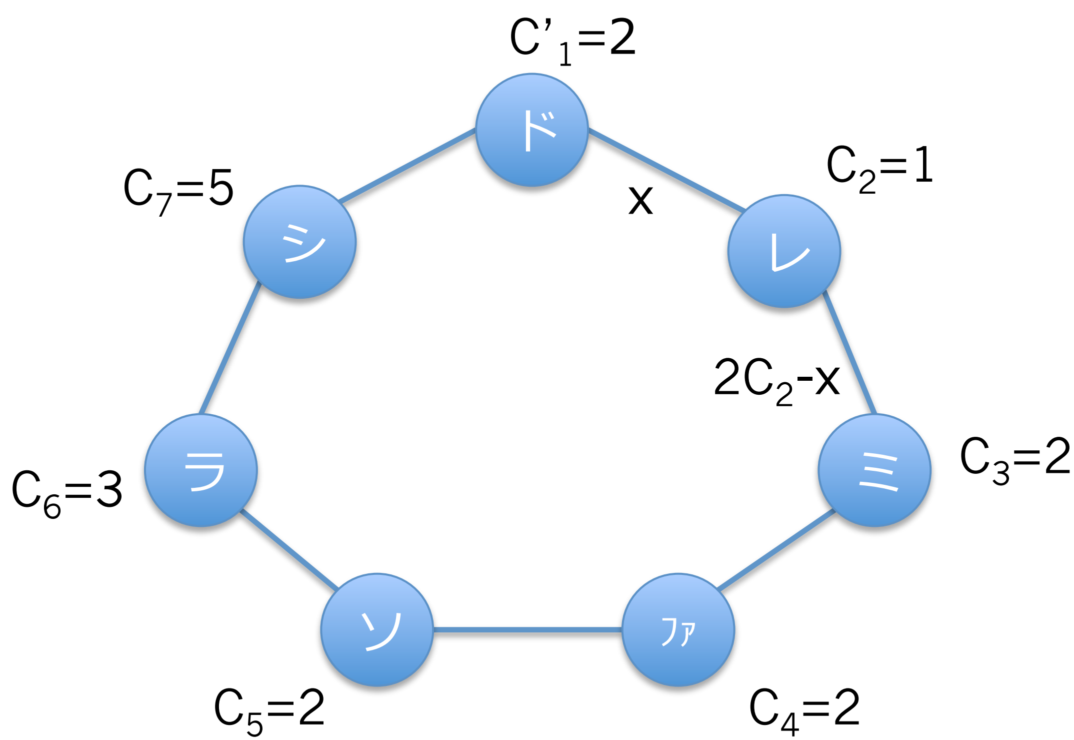
解法

- 辺を通る回数を計算する
  - ド-レの辺を通る回数をxとする
  - すると、他の辺を通る回数はxの一次式となる



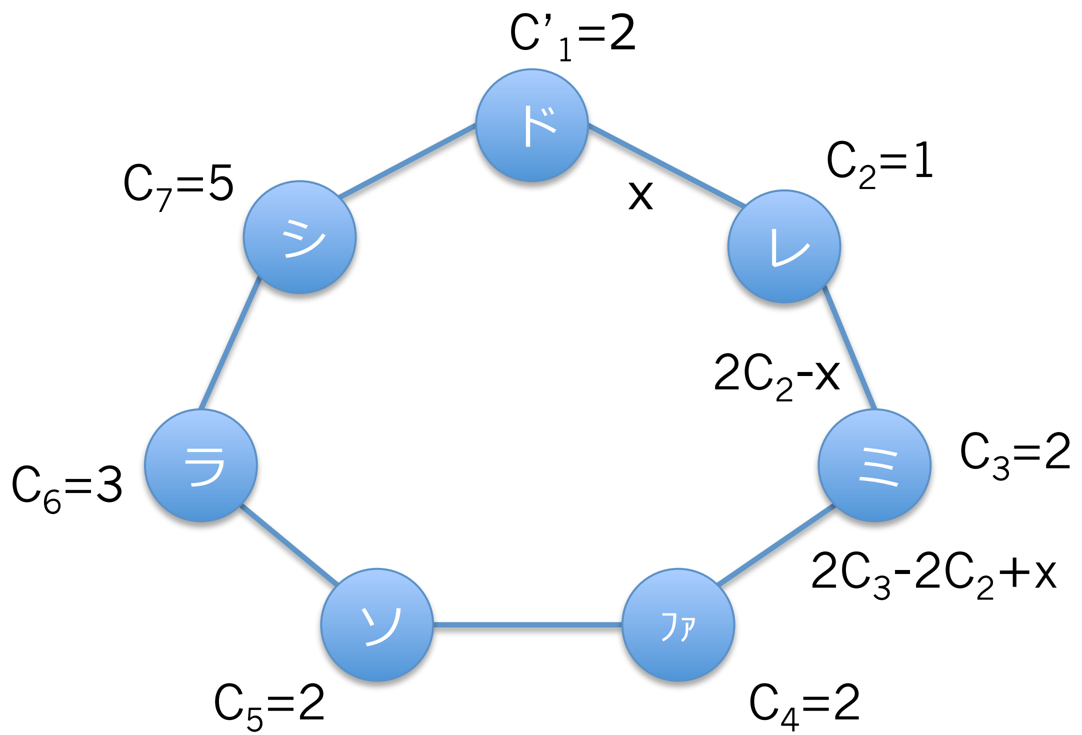
解法

- 辺を通る回数を計算する
  - ドｰレの辺を通る回数をxとする
  - すると、他の辺を通る回数はxの一次式となる



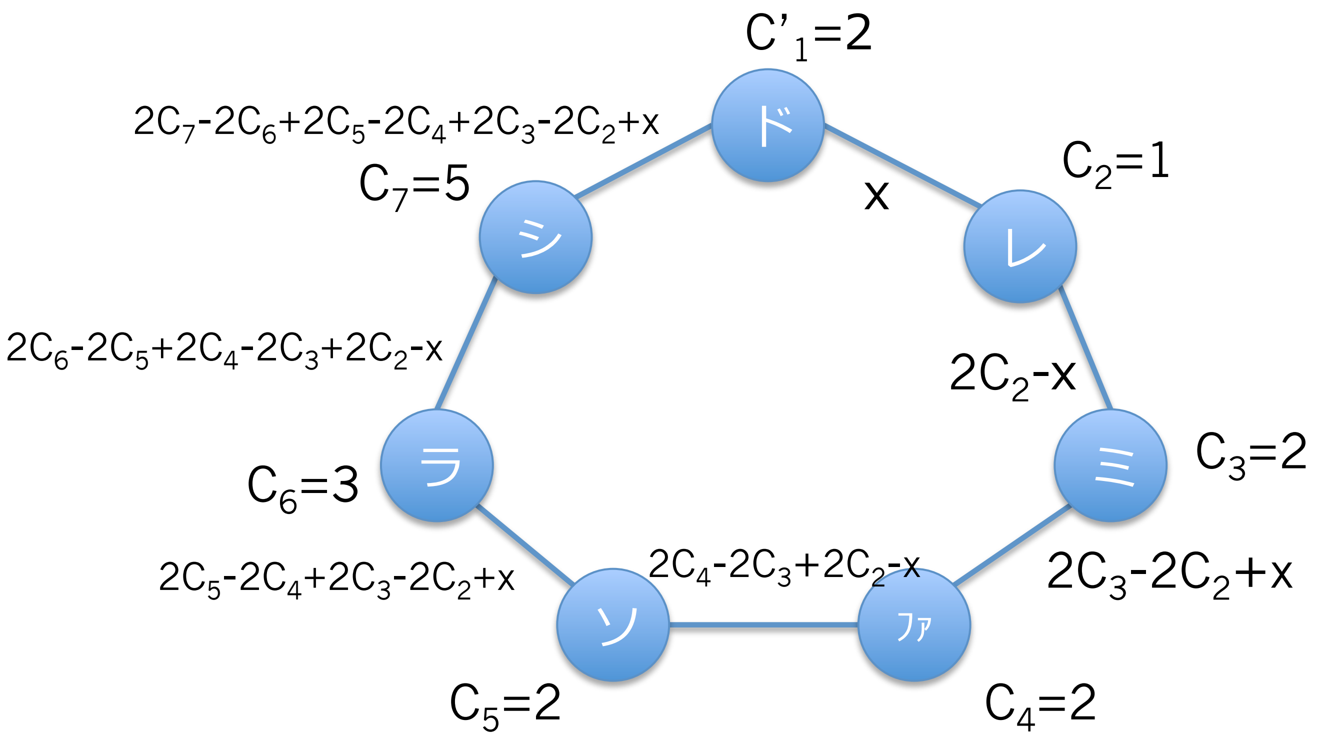
解法

- 辺を通る回数を計算する
  - ド-レの辺を通る回数をxとする
  - すると、他の辺を通る回数はxの一次式となる



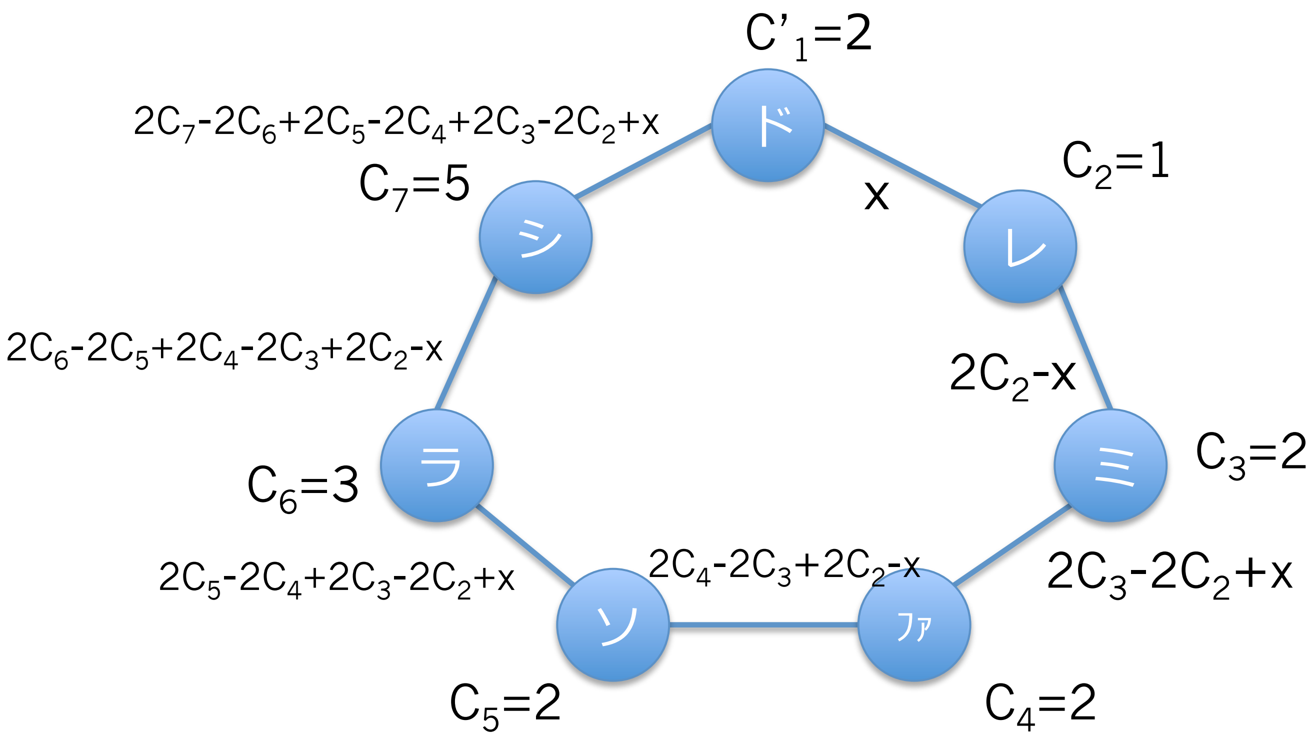
解法

- 辺を通る回数を計算する
  - ド-レの辺を通る回数をxとする
  - すると、他の辺を通る回数はxの一次式となる



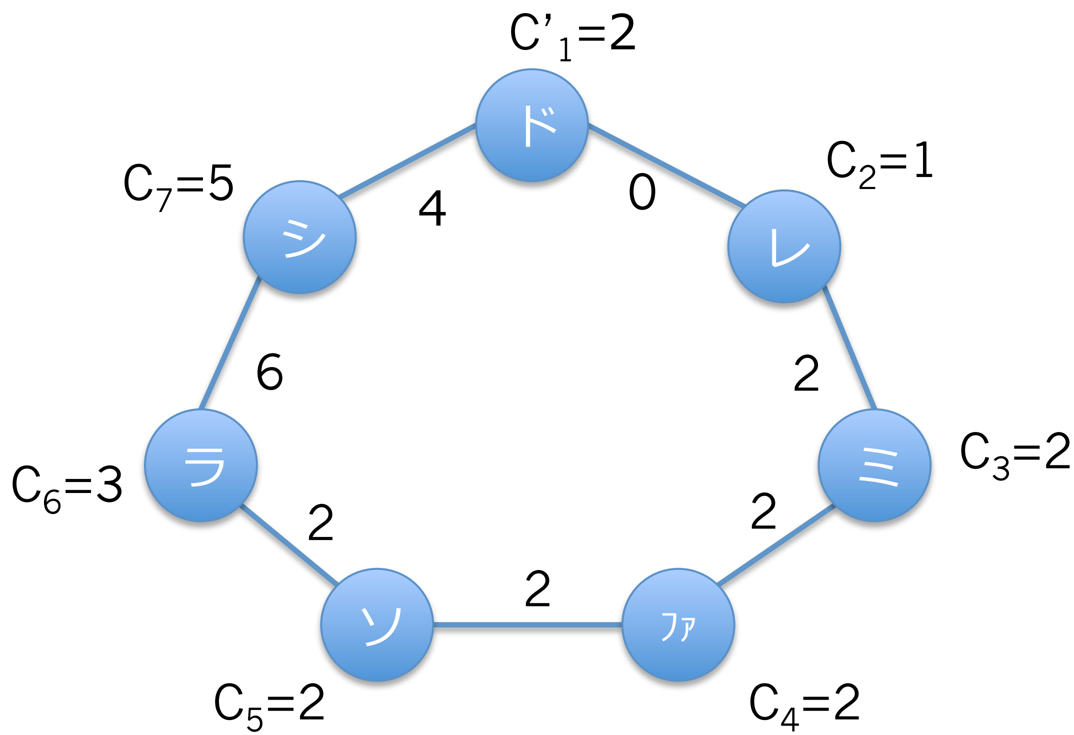
解法

- 辺を通る回数を計算する
  - $(2C_7-2C_6+2C_5-2C_4+2C_3-2C_2+x)+x = 2C'_1$
  - という式が導き出されるので、これを計算すると
  - $x = C'_1-C_7+C_6-C_5+C_4-C_3+C_2$
  - となり、xが求まる



解法

- 各辺を通る回数が求ったあと、どうやって判定するか？
  - 通る回数が負の辺があってはいけない
  - ドを含むような**オイラー路**がなければならない



解法

- オイラー路の存在判定
  - 各頂点の次数が偶数
    - この条件は、「各頂点を踏む回数*2 = 隣接した2つの辺の通る回数の和」として各辺を通る回数を計算したので、常に満たす
  - 次数が0以上の頂点が連結である
    - こっちの判定は必要
    - バグに注意して実装しましょう

G問題 解説
「スタンプラリー」

CODE FESTIVAL 2015 本戦

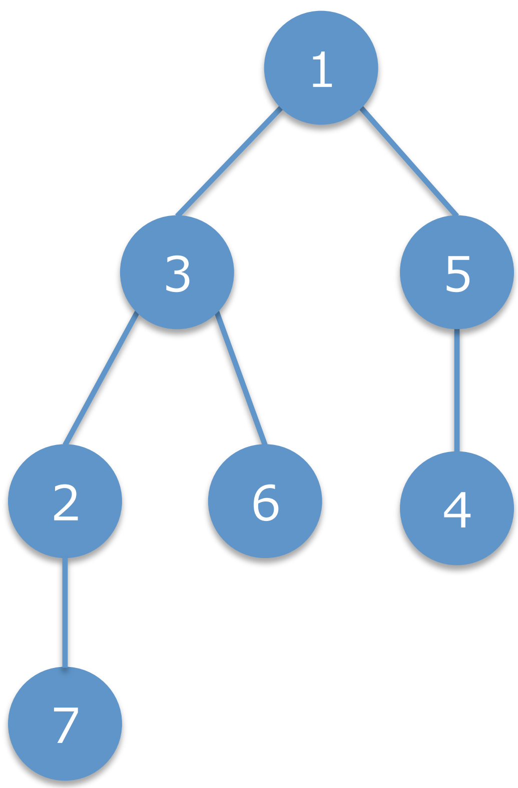
問題概要

- 頂点にラベルのついた根付き木の行きがけ順のうち、辞書順がもっとも小さいものが C1,C2...CN であった
- 元の根付き木として考えられるものは何通りあるか？

- 2 ≦ N ≦ 256

解法

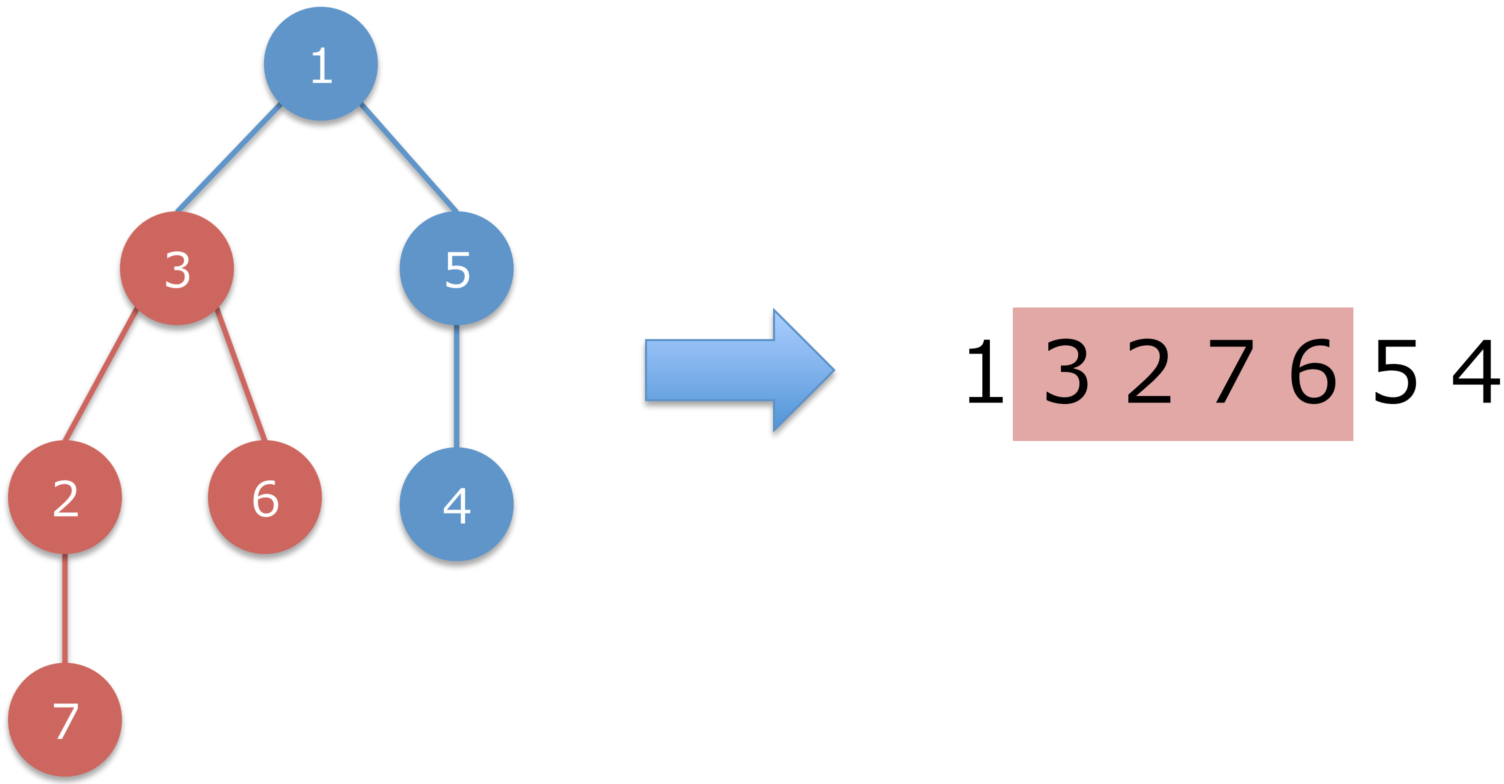
- 行きがけ順の性質



1 3 2 7 6 5 4

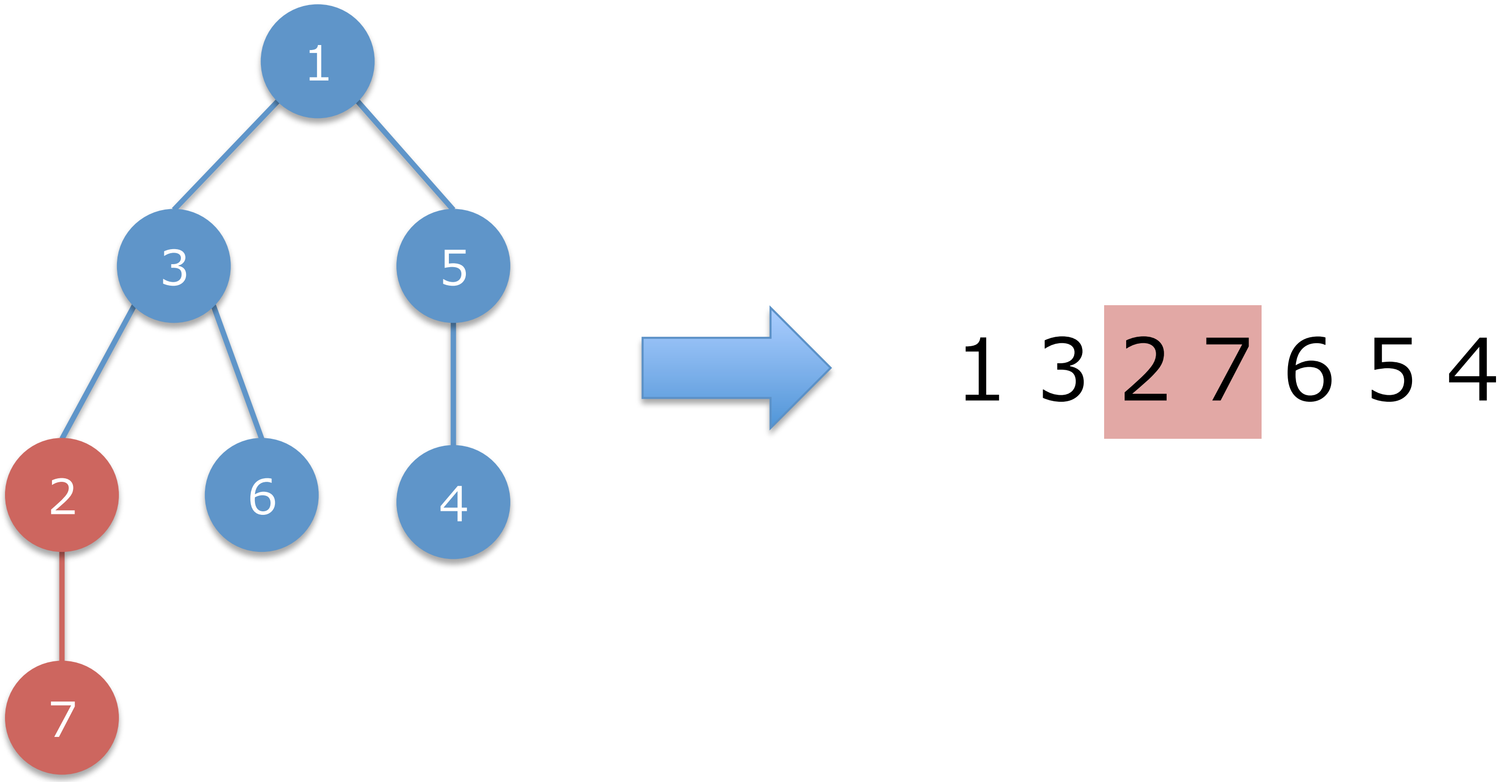
解法

- 行きがけ順の性質
  - 部分木が区間に対応している
  - さらに、区間の先頭は部分木の根



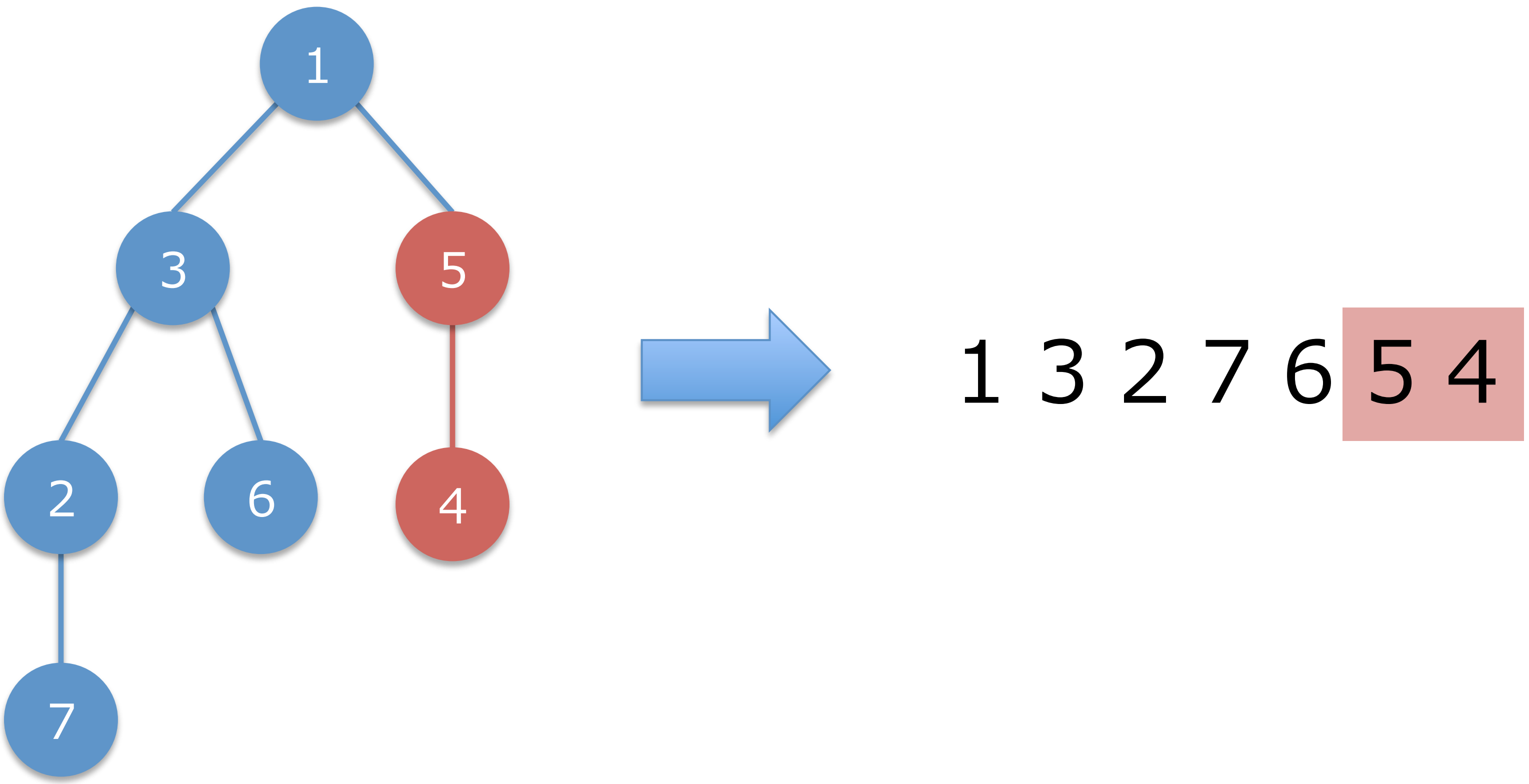
解法

- 行きがけ順の性質
  - 部分木が区間に対応している
  - さらに、区間の先頭は部分木の根



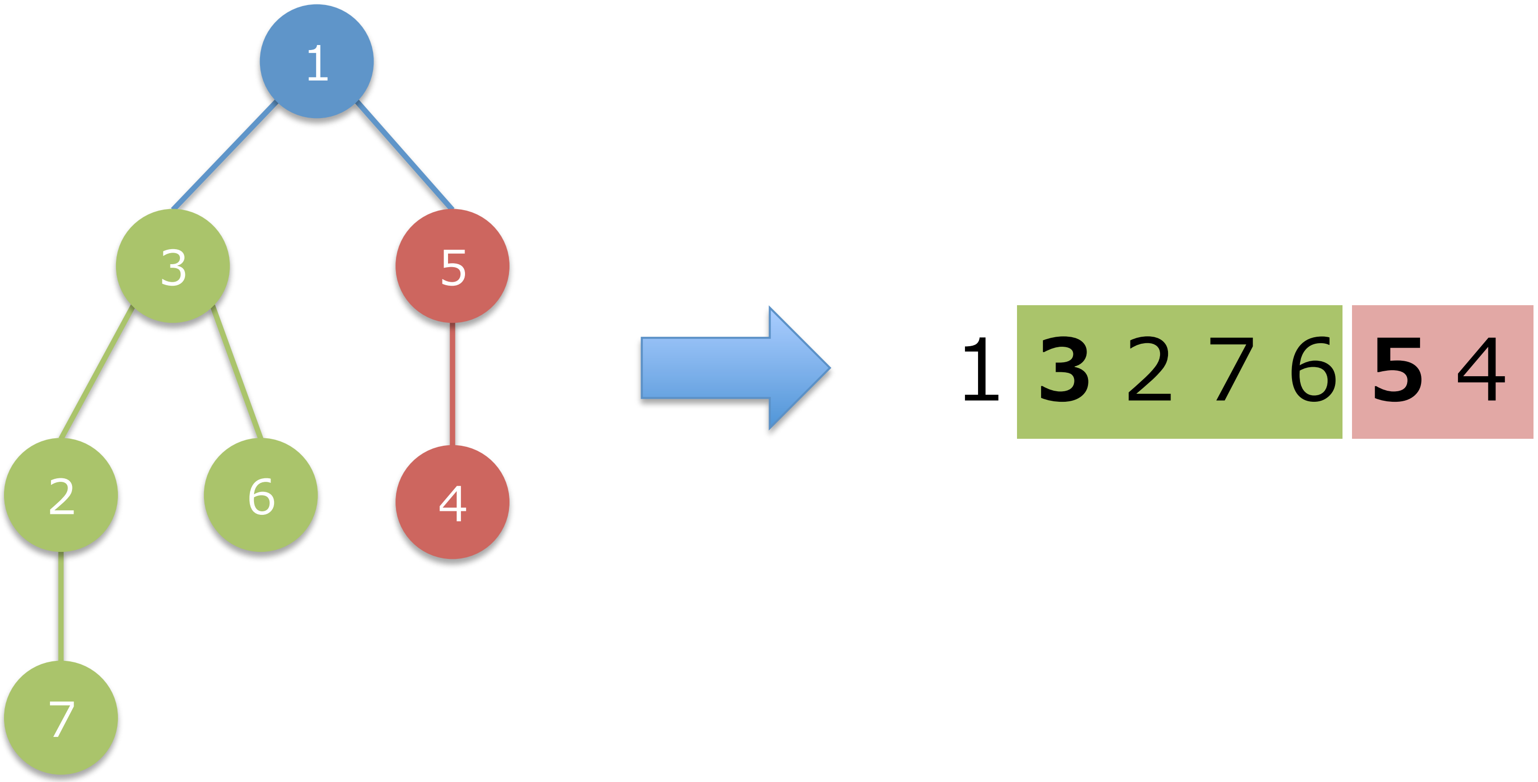
解法

- 行きがけ順の性質
  - 部分木が区間に対応している
  - さらに、区間の先頭は部分木の根



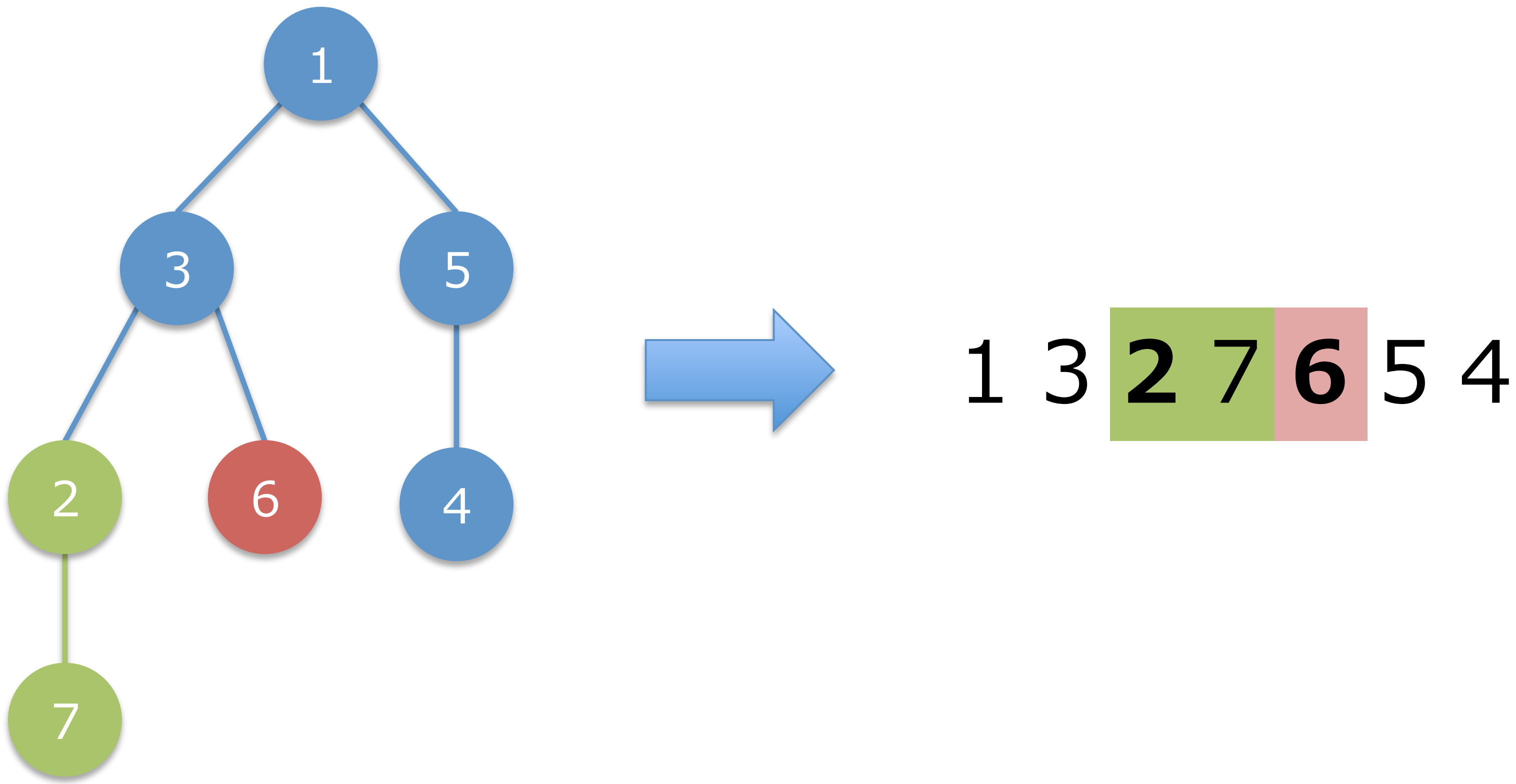
解法

- 辞書順最小の行きがけ順が満たすべき性質
  - 各頂点の子の区間は、先頭の小さい順に並んでいる



解法

- 辞書順最小の行きがけ順が満たすべき性質
  - 各頂点の子の区間は、先頭の小さい順に並んでいる



解法

- 区間DPを計算する
- 状態
  - DP_tree[L][R]
    - C_L～C_Rからなる木（C_Lが根となる）が何通りあるか
  - DP_forest[L][R]
    - C_L～C_Rからなる森（C_Lが最も左の木の根となる）が何通りあるか
- 遷移
  - DP_tree[L][R] = DP_forest[L+1][R]
    - 森に根をくっつける
  - DP_forest[L][R] = 以下のmin
    - min(DP_tree[L][i-1]*DP_forest[i][R])
      - $L < i \leqq R$ かつ $C_L < C_i$ を満たすような i について
      - 森の左に木をくっつける
    - DP_tree[L][R]
      - 木は森である

解法

- 区間DPを計算する
- 初期化
  - DP_tree[i][i] = 1
- 計算量
  - $O(N^3)$

H問題 解説
「焼肉の達人」

CODE FESTIVAL 2015 本戦

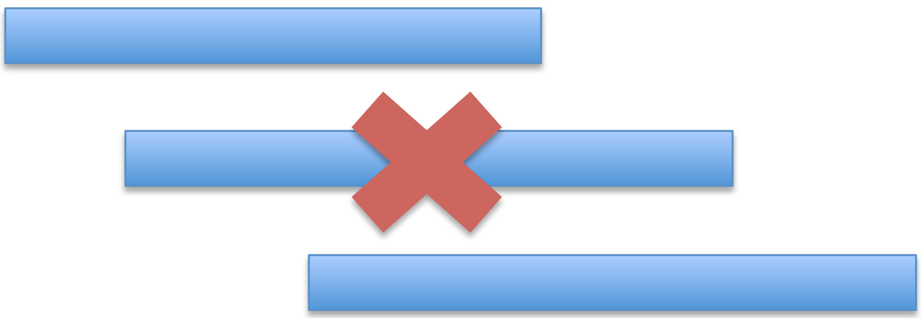
問題概要

- 区間がN個与えられる
- 区間はすべて[0,M]の区間に収まっている
- いくつかの区間を選んで「ちょうど１つの区間だけに覆われている部分」の長さの和を最大化せよ

- 1 ≦ N ≦ 10^5
- 1 ≦ M ≦ 10^5

解法

- 重要な考察
  - 区間が３つ以上重なっている部分ができるように選ぶのは無駄
  - そこに重なっている区間のうち、
    - 左端が最小のもの
    - 右端が最大のもの
  - の２つ（または１つ）以外は選ぶだけ損



解法

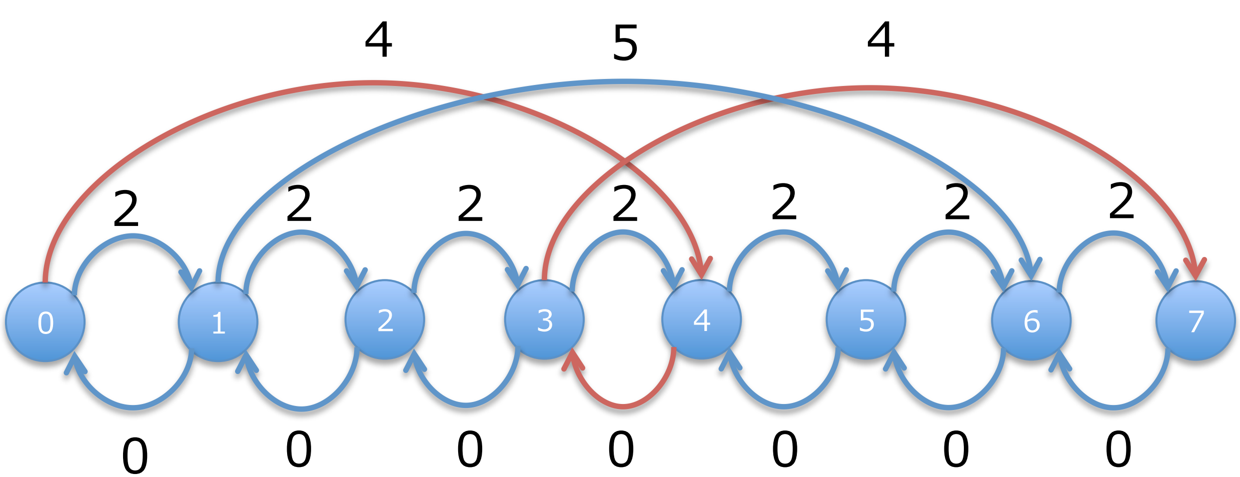
- 各部分に重なっている区間の個数は０か１か２
  - これが１になっている部分の長さの和を最大化したい
- 以下のようにペナルティを設定してみる
  - 重なっている区間の個数が０か２：ペナルティ２
  - 重なっている区間の個数が１：ペナルティ１
- このとき「2M - ペナルティの合計」が重なっている区間が１になっている部分の長さの和となる
- つまり、ペナルティを最小化する問題が解ければ良い

解法

- ペナルティの最小化は以下のような問題に帰着できる
  - 以下の２種類の区間からいくつかの区間を選んで[0,M]の区間を全て覆うとき、最小のコストは？
    - 元の問題で与えられた区間（コスト＝区間の長さ）
    - [x,x+1]の区間（コスト＝２）（合計M個）

解法

- アルゴリズム
  - 以下のように辺を張ったグラフで頂点0から頂点Mへの最短経路をダイクストラ法で求める
    - 頂点は各座標（0~M）
    - 各区間（前のスライドに書いたもの）について、左端→右端へコストが[区間の長さ]の辺を張る
    - 各xについて、x → x-1 へコストが 0 の辺を張る



I問題 解説
「風船ツリー」

CODE FESTIVAL 2015 本戦

問題概要

- 辺に長さのついた頂点数Nの根付き木が与えられる
- ある頂点の高さは根までの辺の長さの和
- 木の高さは頂点の高さの最大値
- Q個の整数が与えられる
- それぞれの整数（Xとする）について、木の高さをちょうどXにするために切る必要のある辺の本数の最小値を求めよ

- 制約
  - $2 \leqq N \leqq 10^5$
  - $1 \leqq Q \leqq 10^5$
  - 1 ≦ 辺の長さ ≦ 10^4

解法

- 辺を切ったときの木の高さとして考えられるものは高々N通りしかない（各頂点の高さ）
- 頂点vを木の中で最も高い頂点にするために切る必要のある辺の本数の最小値をそれぞれの頂点についてもとめればいい

解法

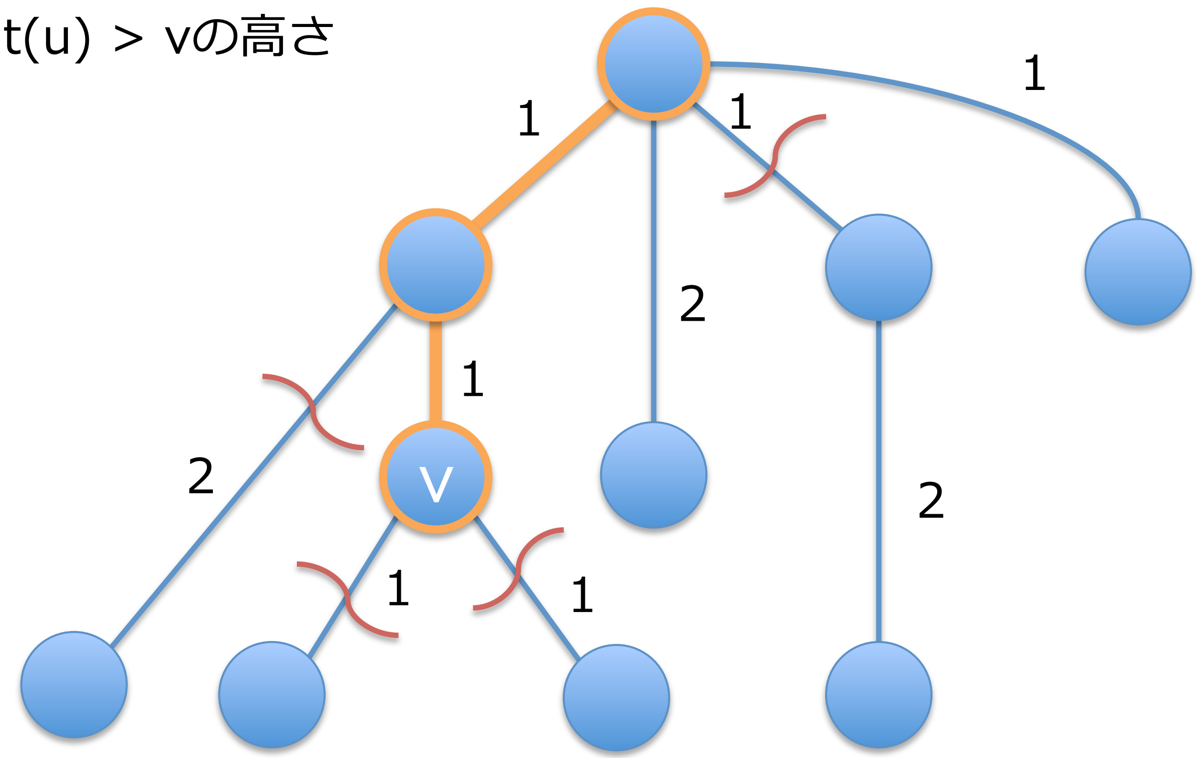
- まず各頂点の高さを求めておく
  - 根から順に計算していけばいい
- このとき、高さの大小関係しか辺を切る本数には影響しないため、圧縮をしておく
  - i番目に大きい高さの値をiに変換する
  - すると、高さが1~NのN種類になる
    - 同じ高さが複数ある場合もう少し少なくなることはある

解法

- さらに、各頂点vについて、子孫のうちの高さの最大値 max_height(v) も求めておく
  - 葉から順に計算していけばいい

解法

- 頂点vを最も高い頂点にするために切る辺は？
  - 根からvまでのパスに含まれる辺は切ってはいけない
  - 根からvまでのパスに含まれる頂点の子(uとする)のうち以下の条件を満たすものを切ると良い
    - vの先祖でない
    - max_height(u) > vの高さ



解法

- アルゴリズム
  - 先祖の直接の子のmax_heightの集合を更新しながら根からDFS
    - 頂点vに訪れた時は、この集合に含まれる数のうちvの高さより大きいものの個数を求めれば良い
  - 集合はBIT(Fenwick tree)で管理すれば良い。
    - 高さを1~Nの値に圧縮しておけば各更新が O(log N)
  - 具体的な更新方法
    - 頂点vに訪れたとき：全ての子のmax_heightを集合に追加
    - 頂点vから子cに潜るとき：cのmax_heightを集合から削除
    - 子cから頂点vに戻ってきたとき：cのmax_heightを集合に追加
    - 頂点vから親に戻るとき：全ての子のmax_heightを集合から削除
  - 計算量は全体で O(N log N) となる

J問題 解説
「N個のバケツ」

CODE FESTIVAL 2015 本戦

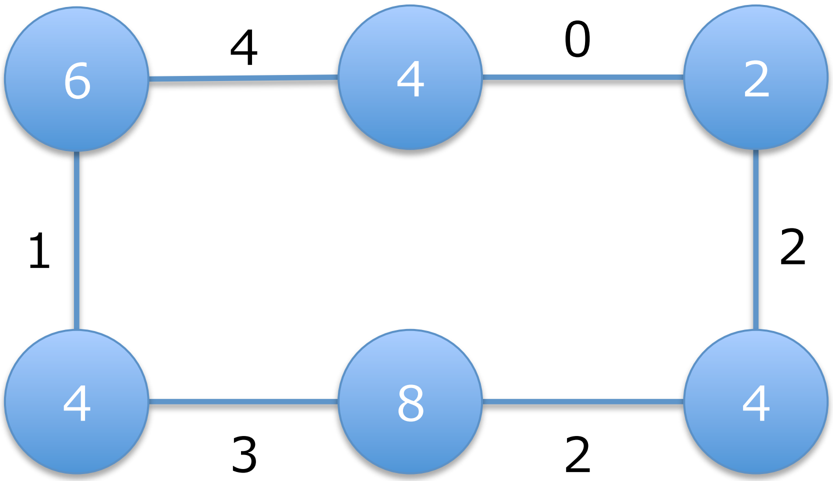
問題概要

- N人のチームメンバーが円周上に並ぶ
- 隣り合った人の間に水を入れたバケツを置く
- 各バケツに入れる水の量は自由に決められる
- バケツを全て持ち上げられれば水の総量がスコアになる
- 各メンバーは、自分の隣にある２つのバケツの水の量の平均が自分の力の値よりも小さければ持ち上げられる
- メンバーが1人変更になるというクエリが Q 個ある
- 各クエリごとに、メンバー変更後のチームで、水の量をうまく選んだ時のスコアの最大値を求めよ

- 制約
  - 3 ≦ N ≦ 10^5、N は偶数（追加点制約ではこの制約はない）
  - 1 ≦ Q ≦ 10^5
  - 1 ≦ メンバー i の力 ≦ 10^4

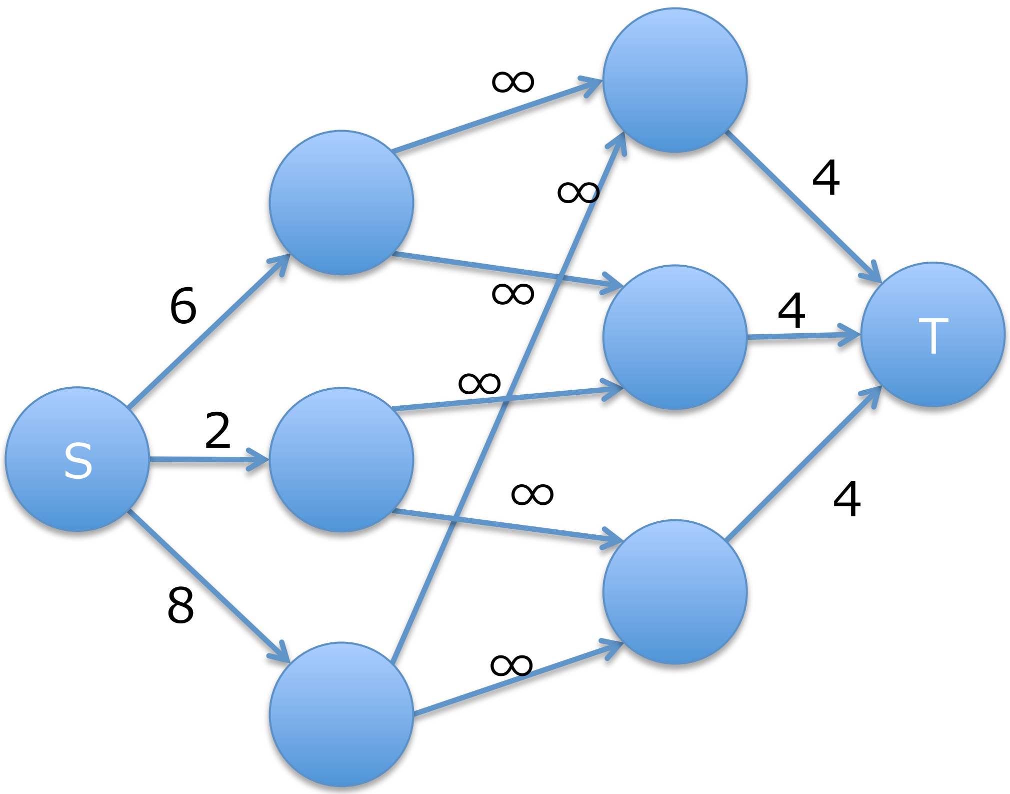
解法

- グラフ問題に帰着してみる
  - 頂点はメンバー、辺はバケツ
  - 辺に 0 以上のコストを設定する
  - 頂点につながっている 2 つの辺のコストの和が頂点に定められた制限（力の値× 2 ）を超えないようにしたい
  - 辺のコストの和の最大値は？



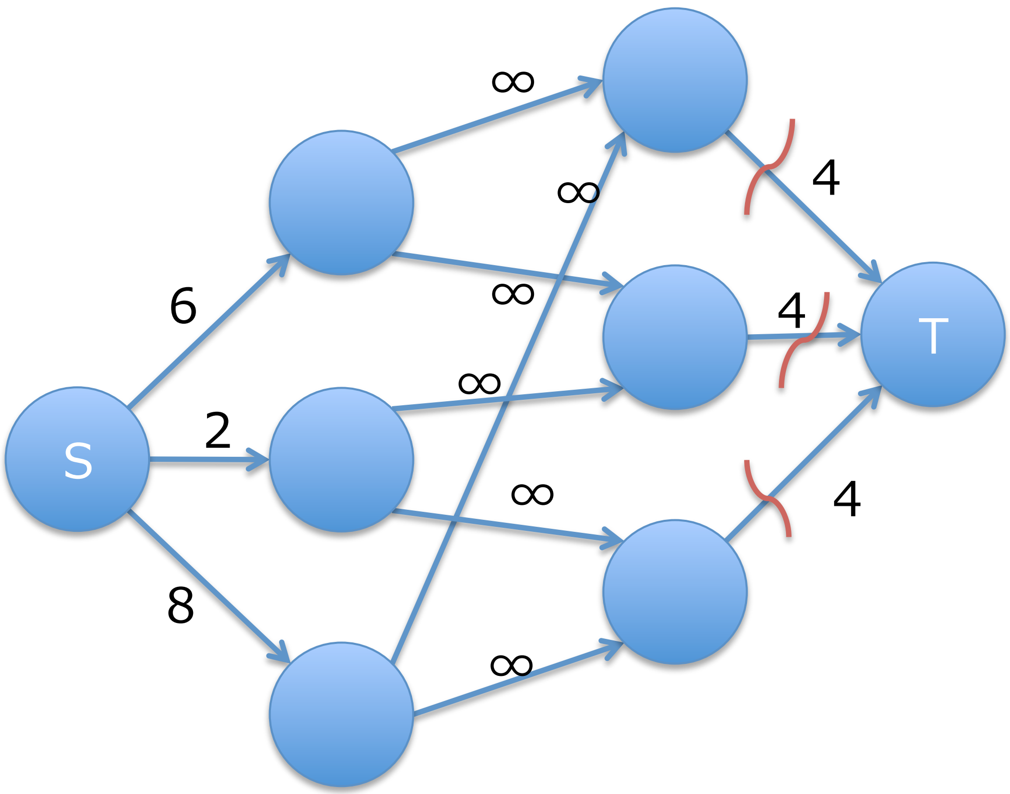
解法

- 最大流問題に帰着してみる
  - メンバー番号の偶奇で左右に分け、ソース/シンクから各メンバーへの辺と、隣り合ったメンバーの間の辺を張る
  - 真ん中の辺の流量が各バケツに入れる水の量に対応する



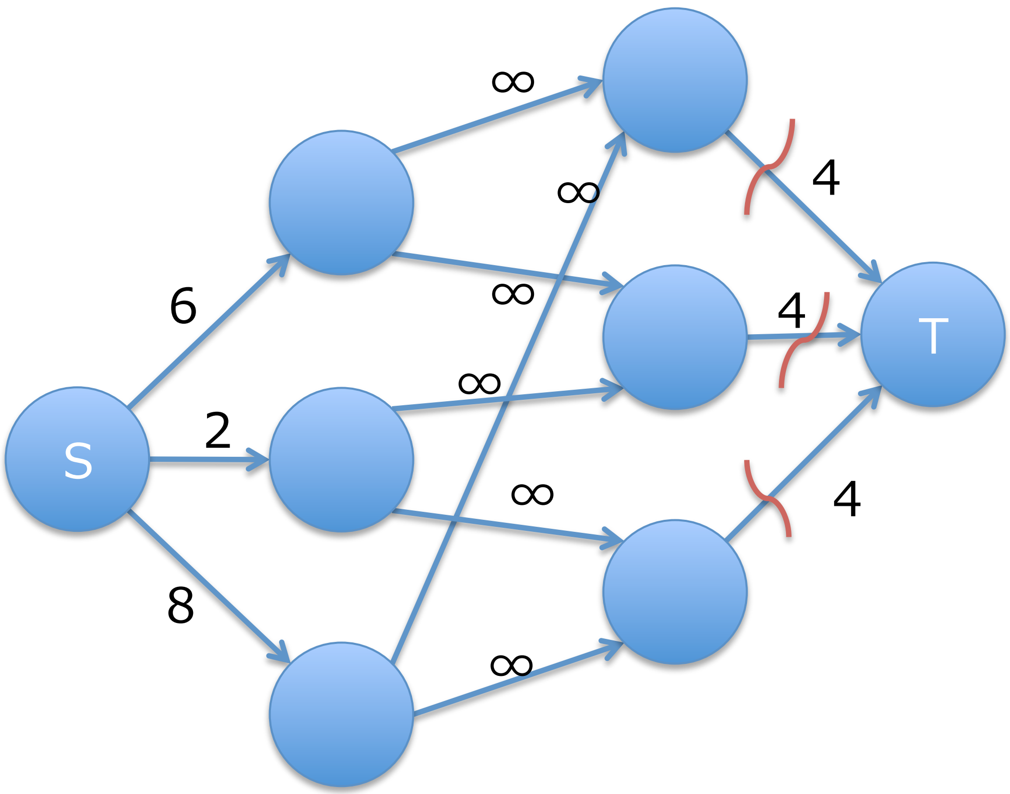
解法

- 最小カット問題に帰着してみる
  - 最大フロー最小カット定理より、最大流＝最小カット



解法

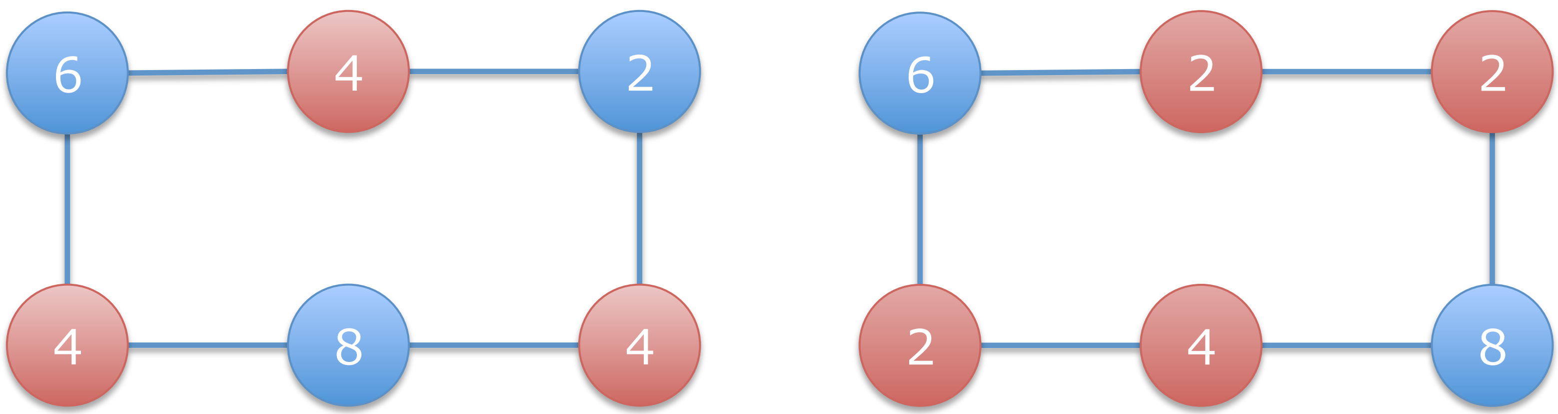
- 最小カットの構造を考察してみる
  - 各バケツごとに、２人のメンバーのうち少なくとも片方の辺を切らなければならない（そうしないとS-Tパスができてしまう）



解法

- つまり？
  - 隣り合った２人のメンバーのうち少なくともどちらか一方が選ばれるように何人かのメンバーを選ぶことを考える
  - このとき選ばれるメンバーの力の和の**最小値**がスコアの最大値

メンバーの力の和が最小となるような選び方の例

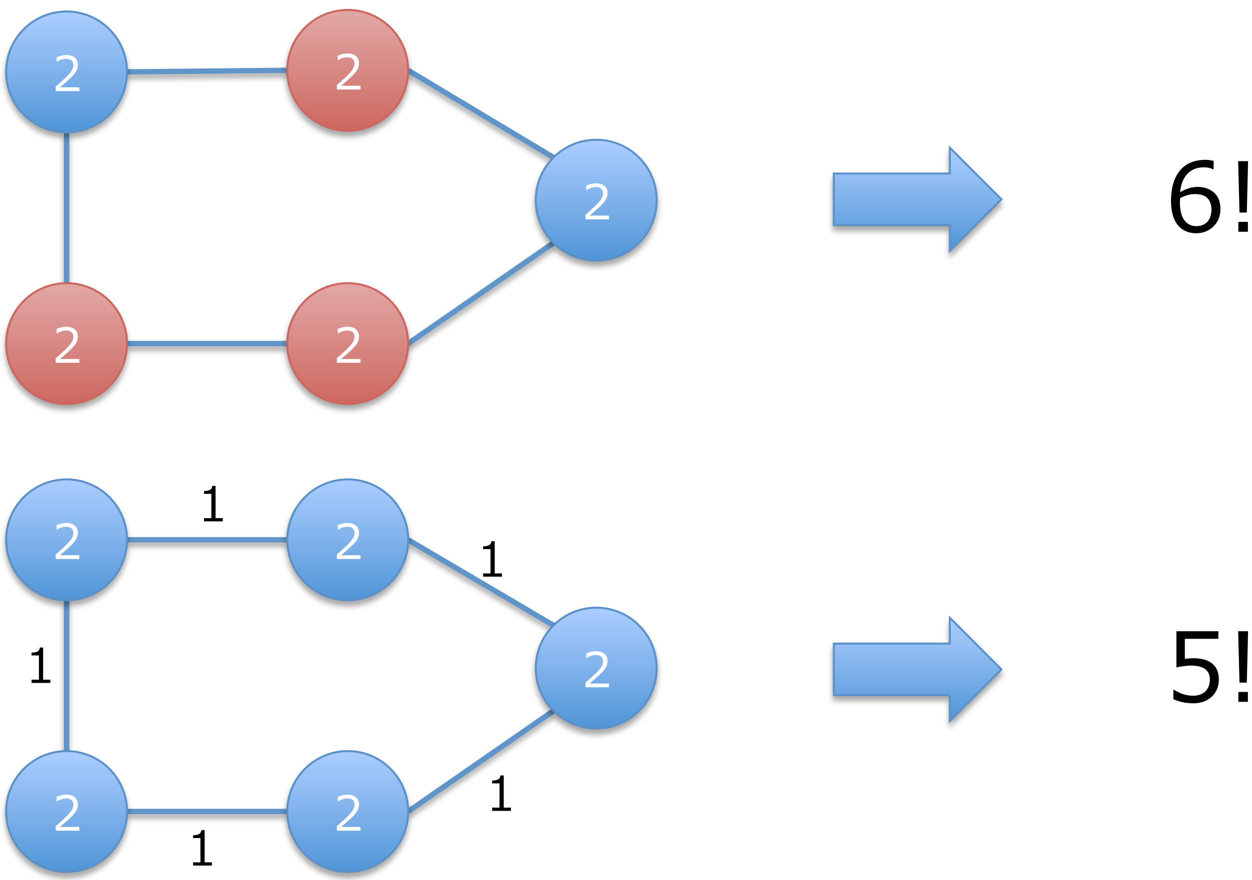


解法

- 以下のデータを計算するsegtreeを用いる
  - ある区間のメンバーの選び方についての以下の４通りの値
    - 左端も右端も選ぶときの力の和の最小値
    - 左端を選び、右端を選ばないときの力の和の最小値
    - 左端を選ばず、右端を選ぶときの力の和の最小値
    - 左端も右端も選ばないときの力の和の最小値
- 1~N の区間についてのデータの、左端も右端も選ばないとき以外の値のうちの最小値が答えとなる
- メンバー更新・答えの計算がともに O(log N) でできる
- 全体の計算量は O(N log N + Q log N) となり、満点を得ることができる

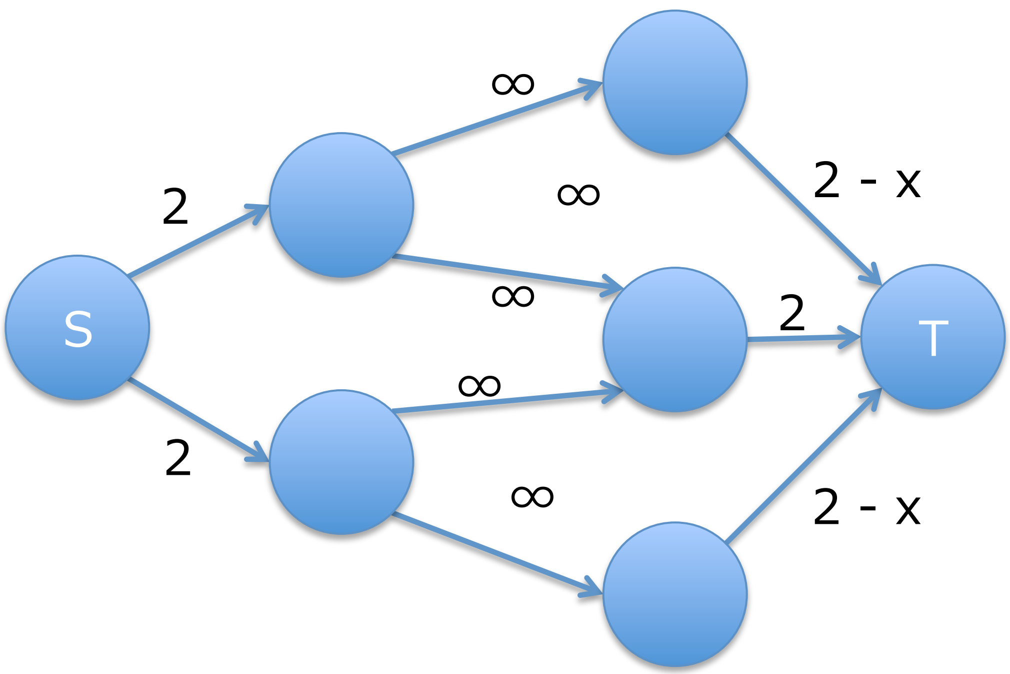
追加点解法

- Nが奇数の場合、先ほどの解法だとダメ



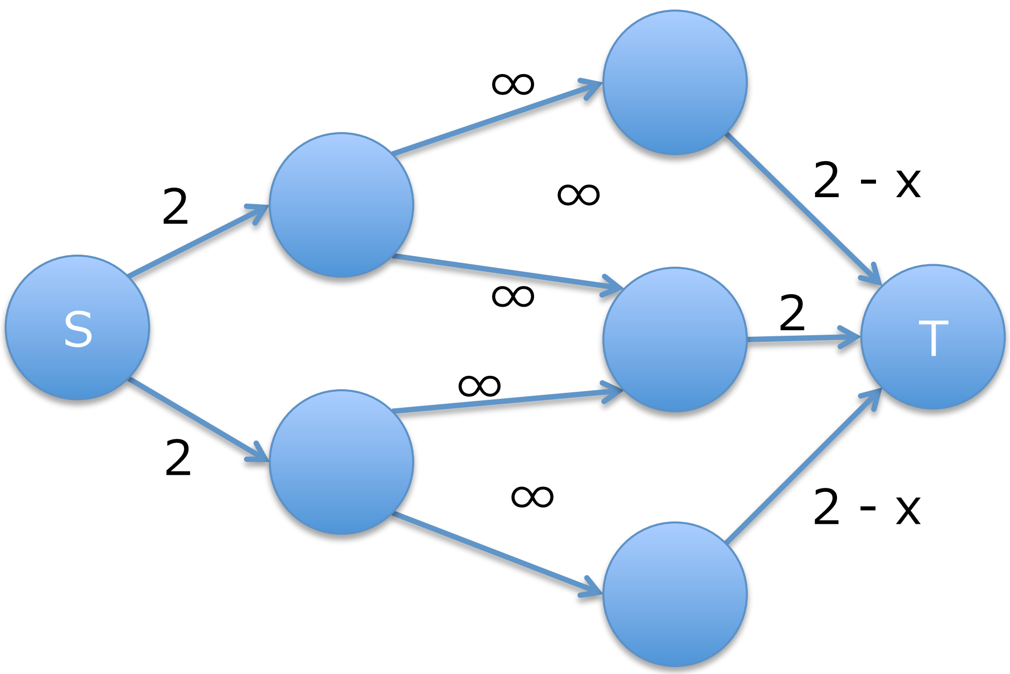
追加点解法

- そもそも最大流のグラフがうまく構築できない
- １ヶ所辺を切って、直線グラフにすると構築できる
  - 切った辺のコストを x とする



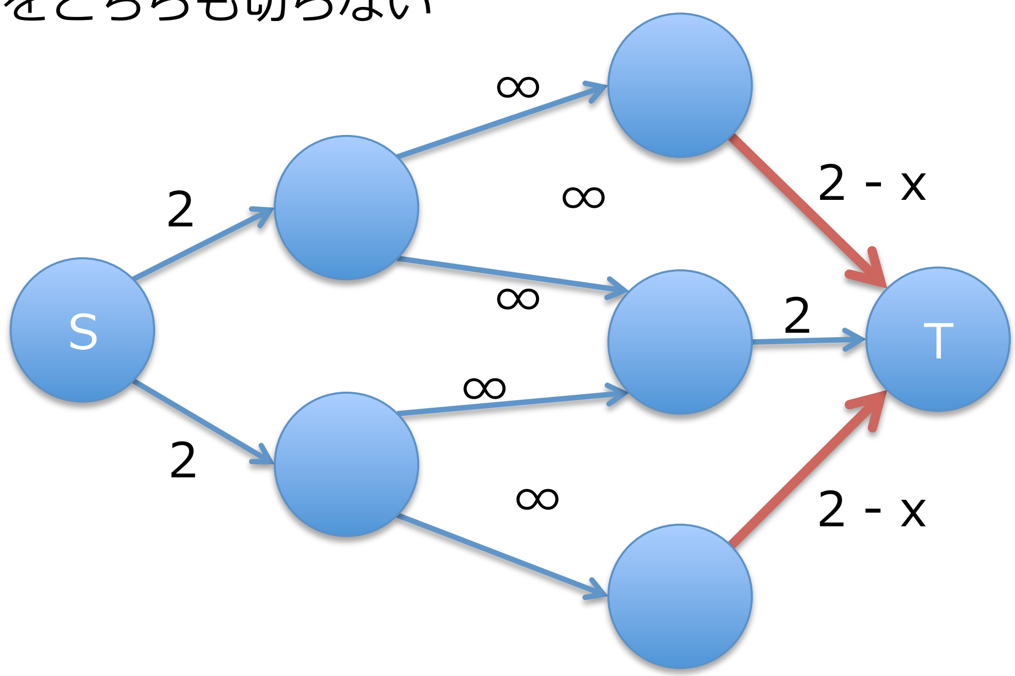
追加点解法

- x を変動させたときの、「mincut+x」の最大値が答え
  - 考えうる x (0~min(両端のメンバーの力)) について全て計算すれば正しい答えは求まりますが、これだとTLE



追加点解法

- mincutの形に注目
  - x を変動させたとき、mincutで切る辺の集合としてありうるものは 3 通りしかない
    - 両端（図の赤い辺）をどちらも切る
    - 両端のうち片方を切る
    - 両端をどちらも切らない



追加点解法

- パターンが切り替わるところの x の値を求める
  - それぞれのパターンの式は以下の通り
    - 両端をどちらも切る：定数 - 2x
    - 両端のうち片方を切る：定数 - x
    - 両端をどちらも切らない：定数
  - 全て x に関しての一次式となる
  - これらのうちの最小値が各 x でのmincutとなる
  - これらのうちの２つの一次式の交点 x 座標（3 通り）と x の上限・下限についてそれぞれmincut+xの値を求め、それらの最大値を求める
- 各クエリごとの計算量を O(log N) に抑えられた
- これで追加点を得ることができる

A問題 解説
「コード川柳」

CODE FESTIVAL 2015 本戦

概要と解説

- 文字列が 3 つ与えられます
- 長さが 5 、 7 、 5 になっているかどうかを判定してください

- まず、文字列を入力します
- 文字列の長さを調べます
- もし間違った長さのものがあれば NO と出力します
- そうでなければ YES と出力します

B問題 解説
「ダイスゲーム」

CODE FESTIVAL 2015 本戦

問題概要

- ６面ダイスをN個振ったとき、和として最も出やすい値を求めよ
- 複数存在する場合はそのうち最も小さい値を求めよ

- $1 \leqq N \leqq 256$

解法

- 確率の分布を考えると、和として出うる値のうちの中央値が最も出やすい
- 7*N/2（小数点以下切捨て）が答え

- ただし、N=1のときは1~6が出る確率が同じなので、答えは1となる

C問題 解説
「寿司タワー」

CODE FESTIVAL 2015 本戦

問題概要

- 寿司をN個を、ネタとシャリ（合計2N個）がある順番で並ぶように積む
- 寿司の積み方は
  - そのまま積む：シャリ、ネタの順に積まれる
  - ひっくり返して積む：ネタ、シャリの順に積まれる
  - 分解して積む：ネタとシャリを別々に積む
- 分解しなければならない寿司の個数の最小値は？

- 制約
  - $1 \leqq N \leqq 256$

解法

- できるだけ分解しない寿司を多くしたい
- 分解せずに積める寿司の個数の最大値を求めたい
- 逆に、ネタとシャリを合体させて寿司を作っていくと考える
- 下から順に見ていき、合体させて寿司が作れる部分を見つけたら寿司を作っていく

シャリ
シャリ
ネタ
ネタ
シャリ
ネタ

解法

- できるだけ分解しない寿司を多くしたい
- 分解せずに積める寿司の個数の最大値を求めたい
- 逆に、ネタとシャリを合体させて寿司を作っていくと考える
- 下から順に見ていき、合体させて寿司が作れる部分を見つけたら寿司を作っていく

シャリ
シャリ
ネタ
ネタ
シャリ
ネタ

解法

- できるだけ分解しない寿司を多くしたい
- 分解せずに積める寿司の個数の最大値を求めたい
- 逆に、ネタとシャリを合体させて寿司を作っていくと考える
- 下から順に見ていき、合体させて寿司が作れる部分を見つけたら寿司を作っていく

シャリ
シャリ
ネタ
ネタ
シャリ
ネタ

解法

- できるだけ分解しない寿司を多くしたい
- 分解せずに積める寿司の個数の最大値を求めたい
- 逆に、ネタとシャリを合体させて寿司を作っていくと考える
- 下から順に見ていき、合体させて寿司が作れる部分を見つけたら寿司を作っていく

シャリ
シャリ
ネタ
ネタ
シャリ
ネタ

2 つ作れた

解法

- 「N - 作ることができた寿司の個数」が答え
- バグを出さないように注意して実装しましょう

D問題 解説
「足ゲームII」

CODE FESTIVAL 2015 本戦

問題概要

- 区間がN個与えられる
- １つだけ区間を取り除くことができる
- 同じ場所に重なっている区間の個数の最大値の最小値を求めよ

- 制約
  - $2 \leqq N \leqq 10^5$
  - $1 \leqq$ 区間の端点の座標 $\leqq 10^5$

解法

- １つも区間を取り除かないときの「同じ場所に重なっている区間の個数の最大値」を m とします
- 区間を１つ取り除いても「同じ場所に重なっている区間の個数の最大値」は高々1しか減らないため、答えは m または m-1 になる
- 区間を１つ取り除いて「同じ場所に重なっている区間の個数の最大値」を減らすことができるかどうかを判定できれば良い

解法

- 「同じ場所に重なっている区間の個数の最大値」を減らすことができるような区間とはどんな区間？
  - m 個の区間が重なっている部分を全て覆っているような区間

解法

- m 個の区間が重なっている部分を全て覆っているような区間の見つけ方
  - まず、各部分をいくつの区間が覆っているかを求める
    - imos法（累積和）などを用いて計算できる
  - 次に m 個の区間が重なっている部分を列挙する
  - そのうち、最も左にあるものと右にあるものを両方覆うことができる区間を探す。

- 計算量は O(N) となる

E問題 解説
「ショートコーディング」

CODE FESTIVAL 2015 本戦

問題概要

- 単項演算子「-」「!」のみからなる式が与えられる
  - 「-」：正負を反転する
  - 「!」：0を1に、それ以外を0にする
- -256~256 の入力についての出力が全て同じになるような式のうち、最も短いものを求めよ
- 1 ≦ 式の長さ ≦ 256

解法

- 考察
  - 「!」の演算結果には入力の正負は関係ない
    - →「!」の後ろにある「-」はあってもなくても変わらない
    - そういうものを消すと「-----!!!!」のように、「-」がx個並んだ後に「!」がy個並んだ式になる

解法

- 考察
  - 「--」は消して良い
    - 「-」がx個並んだ式は「-」がx%2個ならんだ式に書き換えてよい

解法

- 考察
  - 「!!」は基本的に消して良い
    - ただし、「!!」と「」は別の結果となるため、残りが２個の場合は消してはいけない
    - つまり「!」がy個ならんだ式は、
      - y=0：「」
      - y>0、yが奇数：「!」
      - y>0、yが偶数：「!!」

解法

- 解法
  - まず、「!」の後ろにある「-」を消す
  - 「--」を消していく
  - 「!」の個数が2以下になるまで「!!」を消していく

- ちなみに、答えは一意に定まります

F問題 解説
「歩くピアニスト」

CODE FESTIVAL 2015 本戦

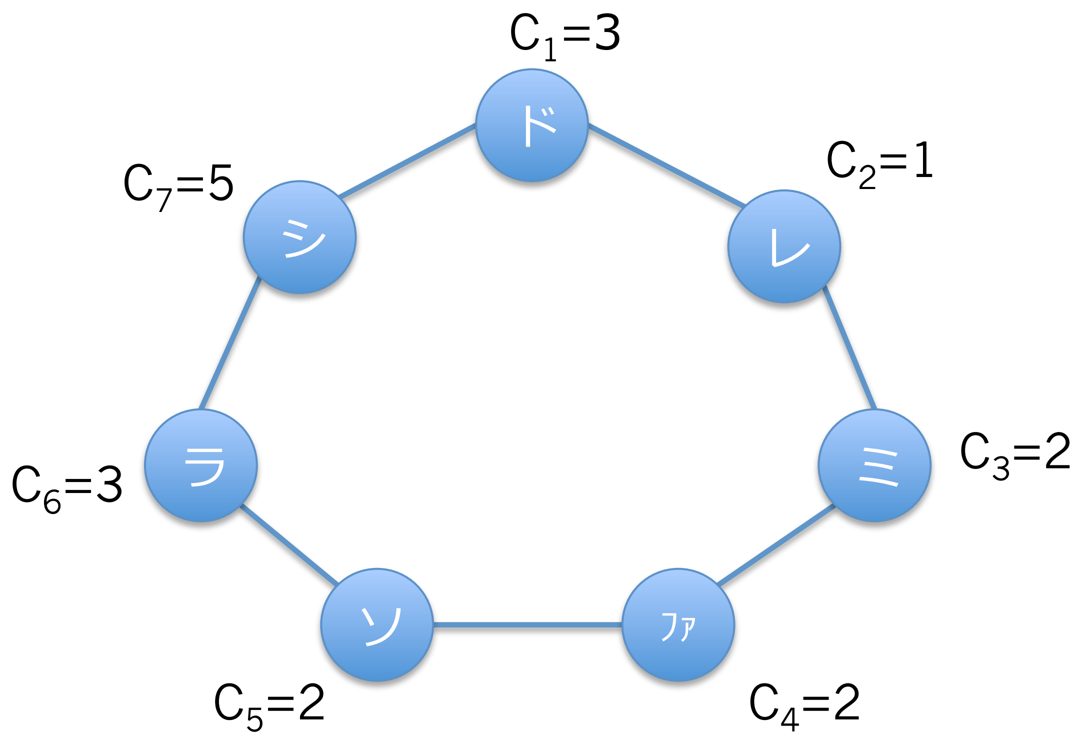
問題概要

- 巨大なピアノがある
- 以下のような演奏ができるかどうかを判定せよ
  - ド～シをそれぞれちょうどC_1~C_7回鳴らす
  - ある音の直後には隣り合った音しか鳴らすことができない
    - 例えばド→レ、ファ→ミ、ド→シなどはOK
  - ドから始めて、ドで終わる

- 制約
  - $0 \leqq C_1 \sim C_7 \leqq 10^{10}$
  - C_1~C_7の和は 1 以上

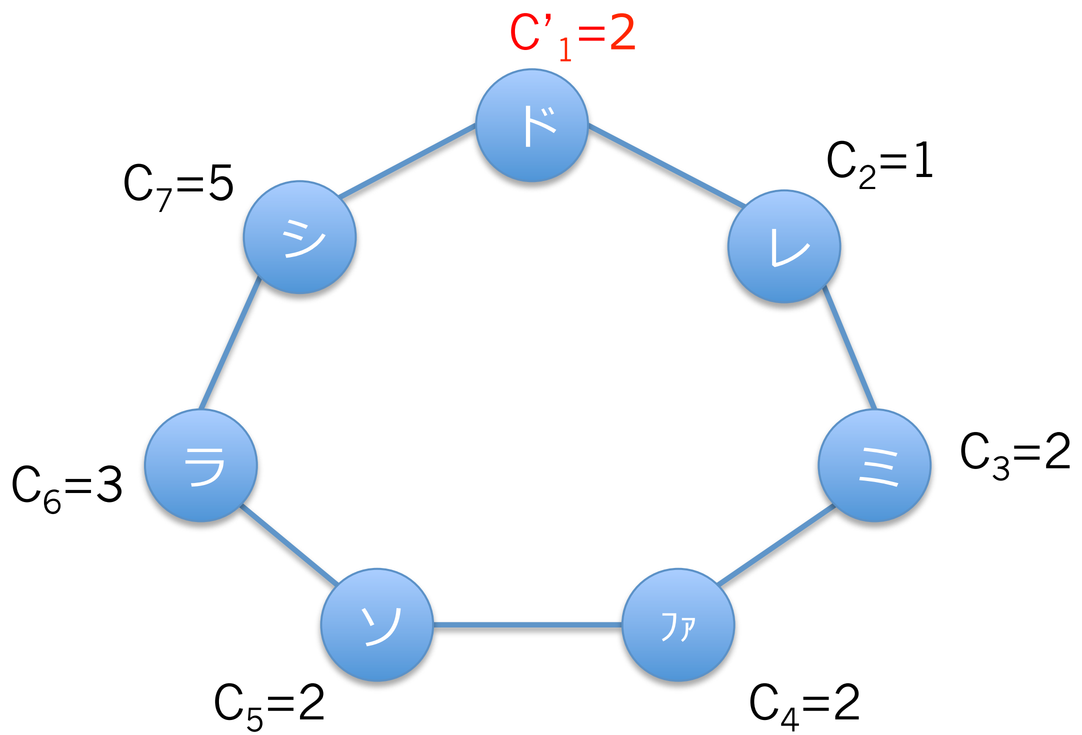
解法

- グラフ問題に帰着する
  - 頂点は音階、辺は隣り合った音を結ぶ
  - ドからはじめてドで終わるようなパスで、各頂点を通る回数がそれぞれちょうどC_1~C_7回になっているようなものがあるか



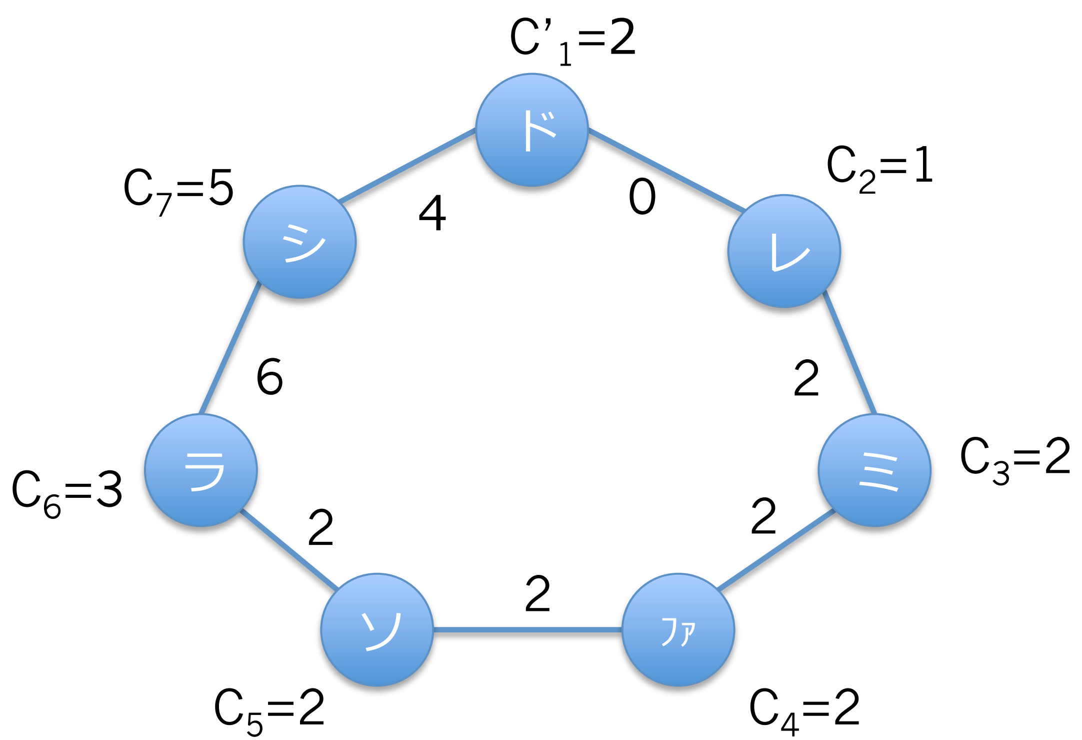
解法

- 計算しやすいように、少し言い換える
  - ドからはじめてドで終わるようなパスで、各頂点を通る回数がそれぞれちょうどC_1~C_7回になっているようなものがあるか
  - →ドを含むようなサイクルで、各頂点を通る回数がそれぞれちょうどC_1-1,C_2~C_7回になっているようなものがあるか



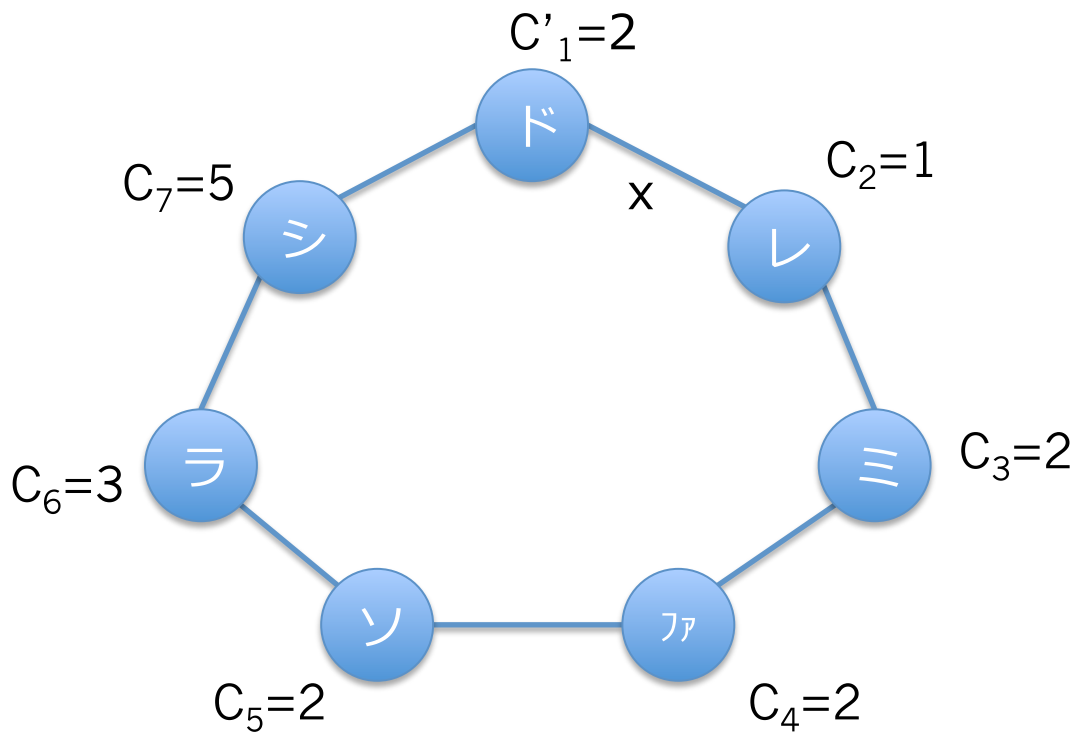
解法

- 辺を通る回数に注目する
  - 辺を通る回数はどういう条件を満たせばいいか
    - 各頂点を踏む回数*2 = 隣接した２つの辺の通る回数の和
      - 各頂点は１回の訪問につき入るときと出るときで２回辺を使うため



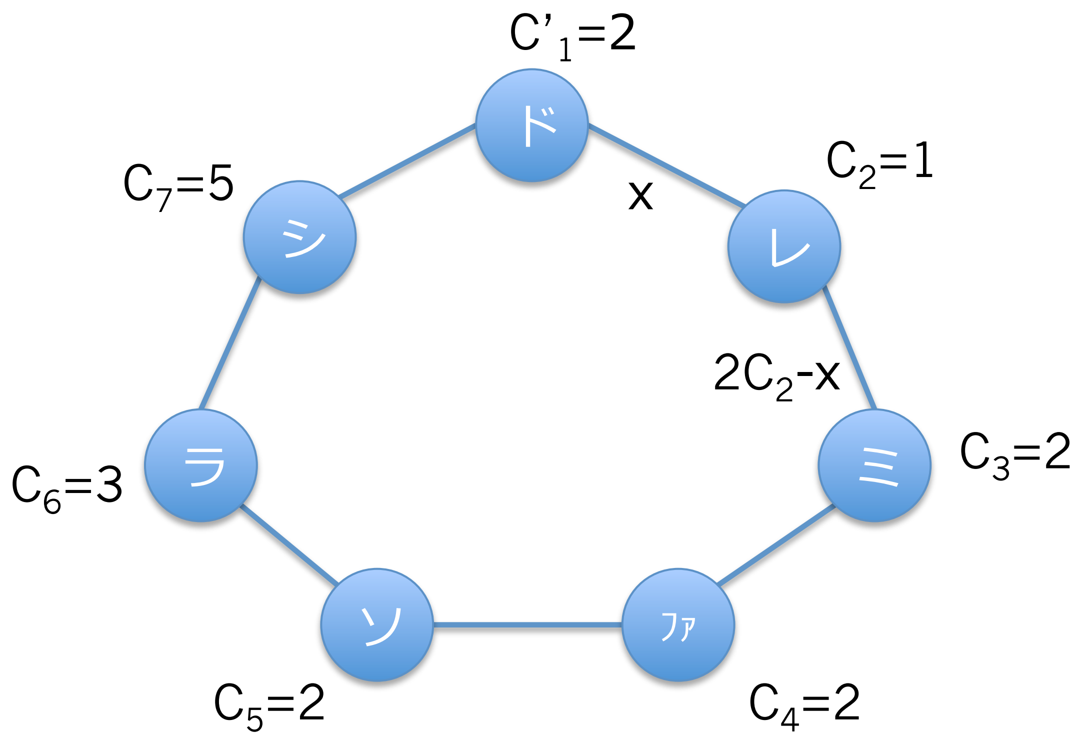
解法

- 辺を通る回数を計算する
  - ド-レの辺を通る回数をxとする
  - すると、他の辺を通る回数はxの一次式となる



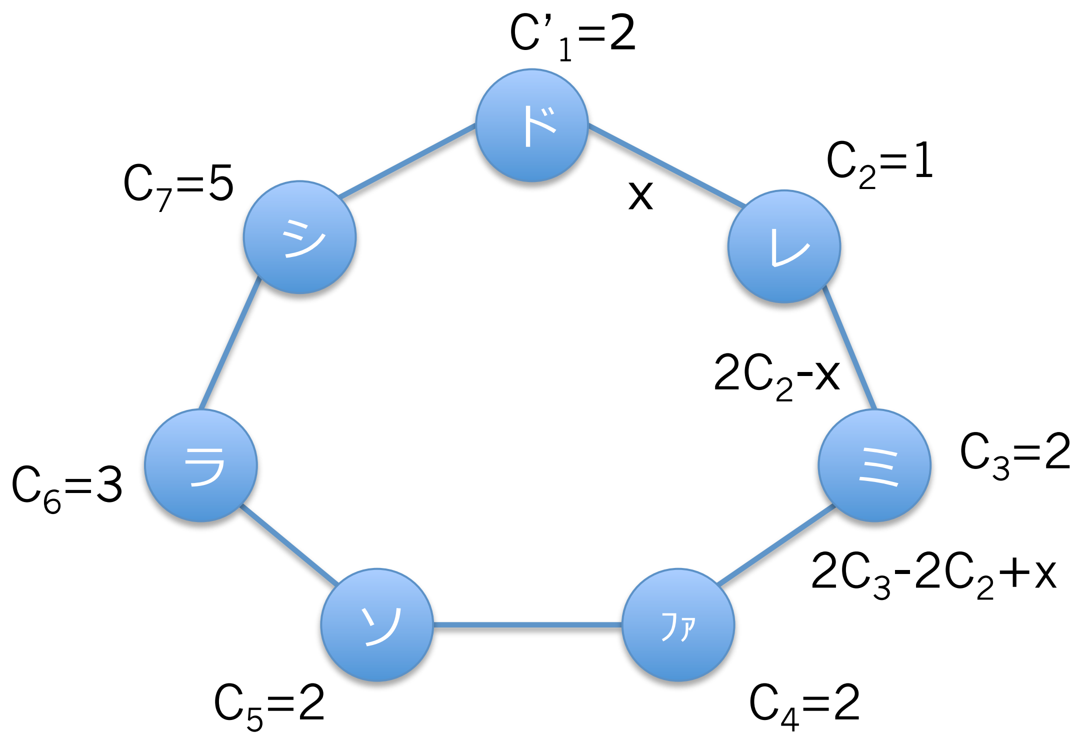
解法

- 辺を通る回数を計算する
  - ド-レの辺を通る回数をxとする
  - すると、他の辺を通る回数はxの一次式となる



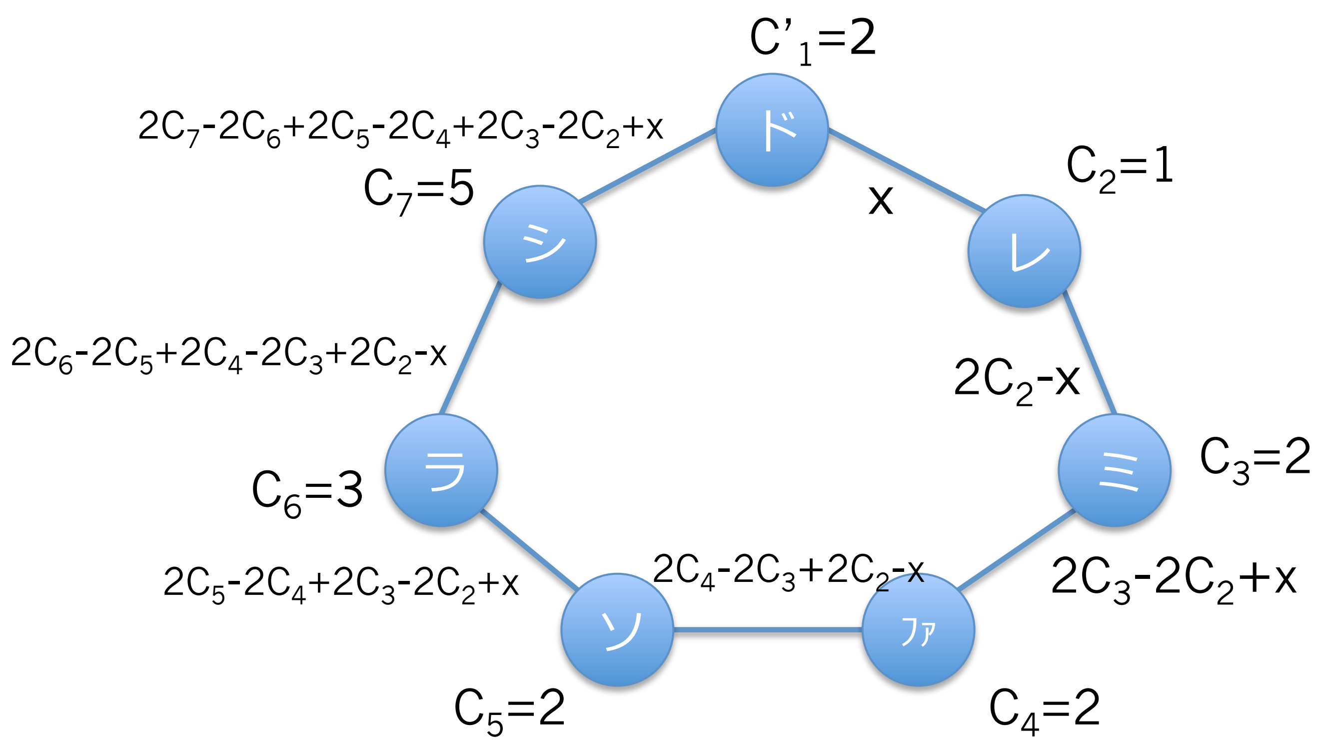
解法

- 辺を通る回数を計算する
  - ド-レの辺を通る回数をxとする
  - すると、他の辺を通る回数はxの一次式となる



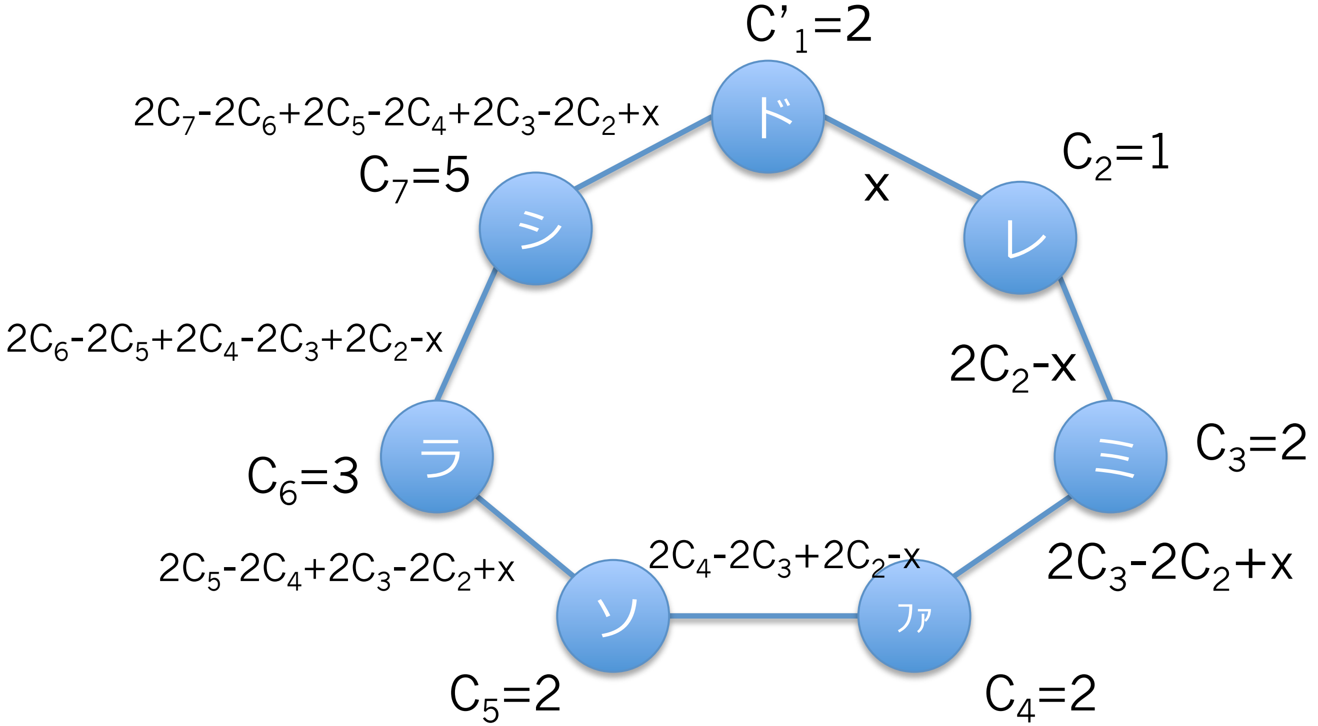
解法

- 辺を通る回数を計算する
  - ド-レの辺を通る回数をxとする
  - すると、他の辺を通る回数はxの一次式となる



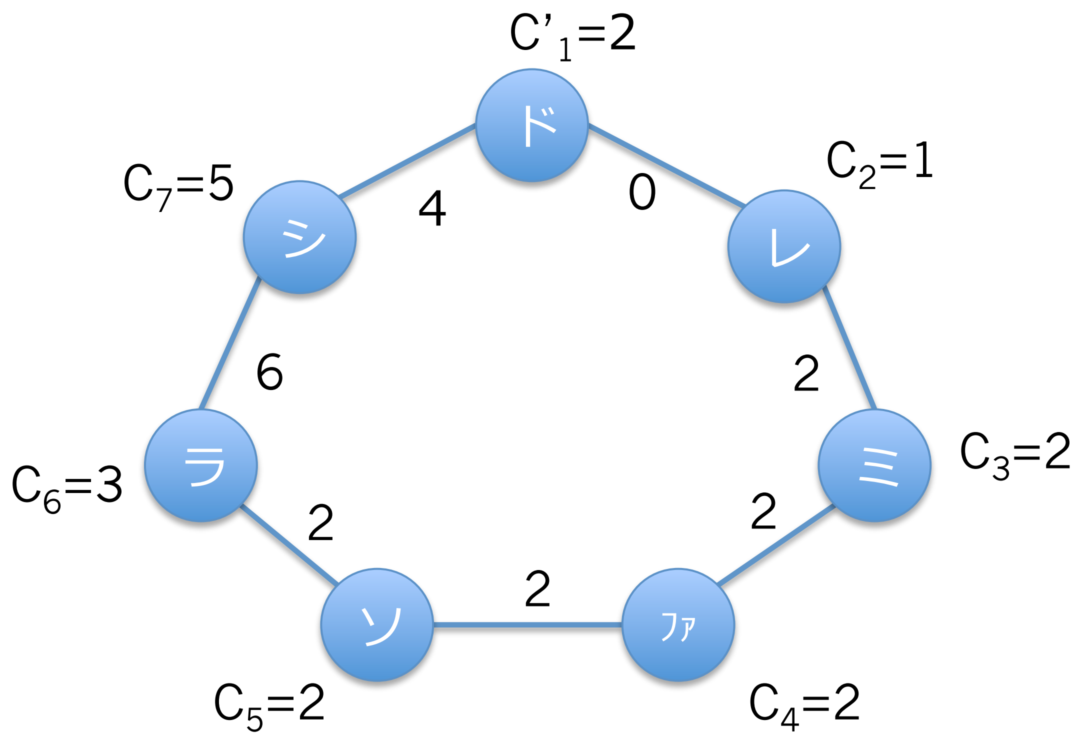
解法

- 辺を通る回数を計算する
  - $(2C_7-2C_6+2C_5-2C_4+2C_3-2C_2+x)+x = 2C'_1$
  - という式が導き出されるので、これを計算すると
  - $x = C'_1-C_7+C_6-C_5+C_4-C_3+C_2$
  - となり、xが求まる



解法

- 各辺を通る回数が求ったあと、どうやって判定するか？
  - 通る回数が負の辺があってはいけない
  - ドを含むような**オイラー路**がなければならない



解法

- オイラー路の存在判定
  - 各頂点の次数が偶数
    - この条件は、「各頂点を踏む回数*2 = 隣接した2つの辺の通る回数の和」として各辺を通る回数を計算したので、常に満たす
  - 次数が0以上の頂点が連結である
    - こっちの判定は必要
    - バグに注意して実装しましょう

G問題 解説
「スタンプラリー」

CODE FESTIVAL 2015 本戦

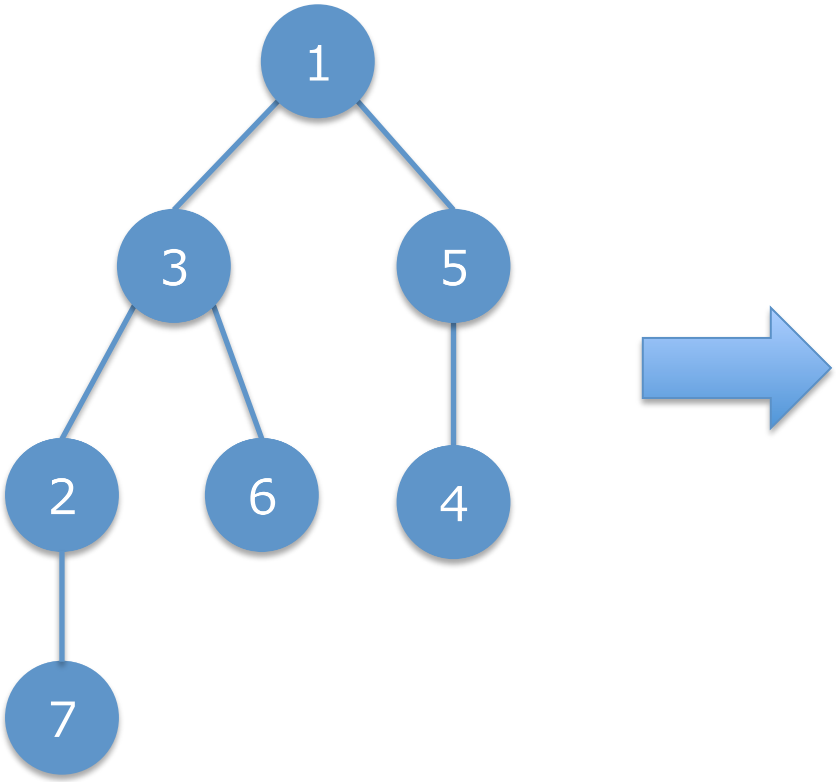
問題概要

- 頂点にラベルのついた根付き木の行きがけ順のうち、辞書順がもっとも小さいものが C1,C2…CN であった
- 元の根付き木として考えられるものは何通りあるか？

- 2 ≦ N ≦ 256

解法

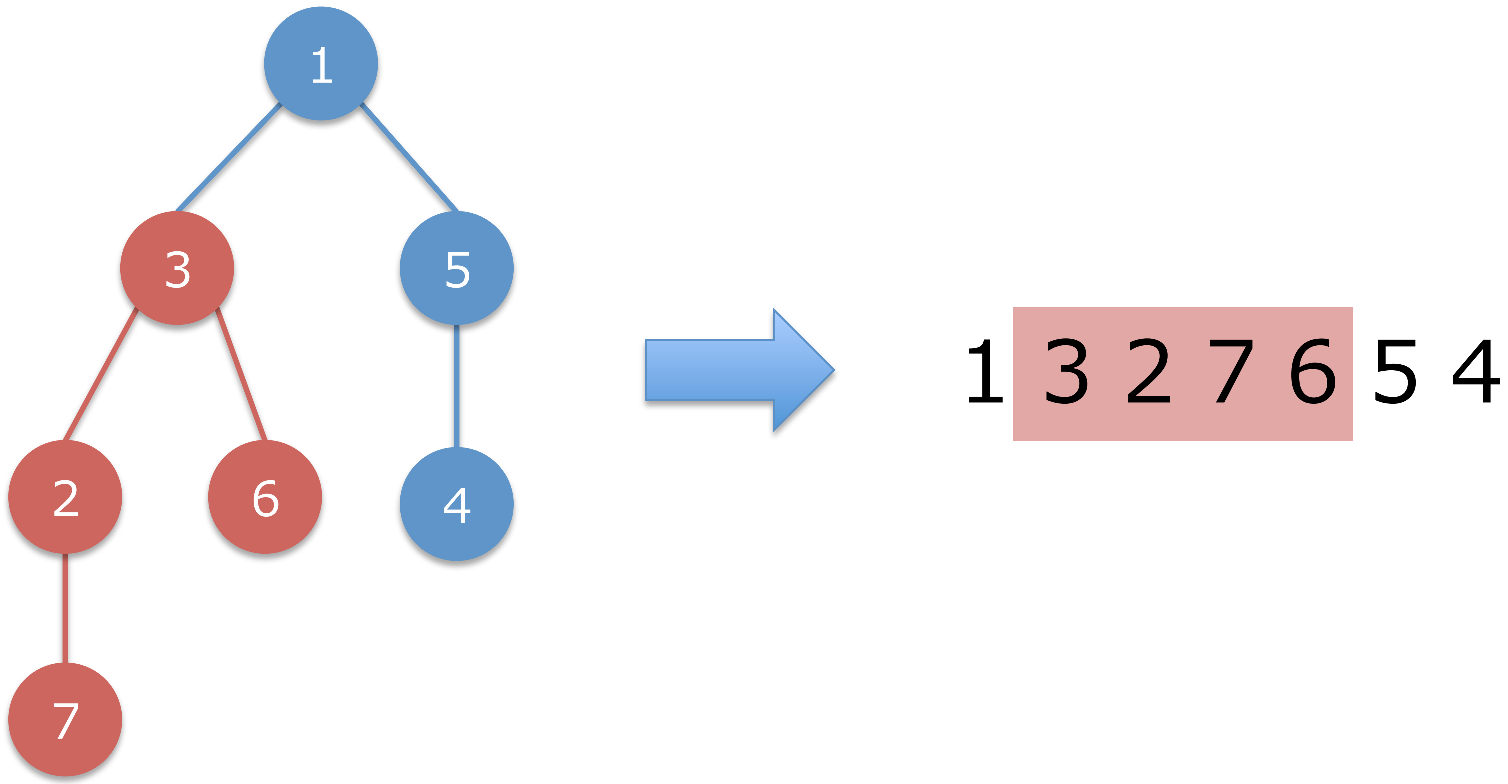
- 行きがけ順の性質



1 3 2 7 6 5 4

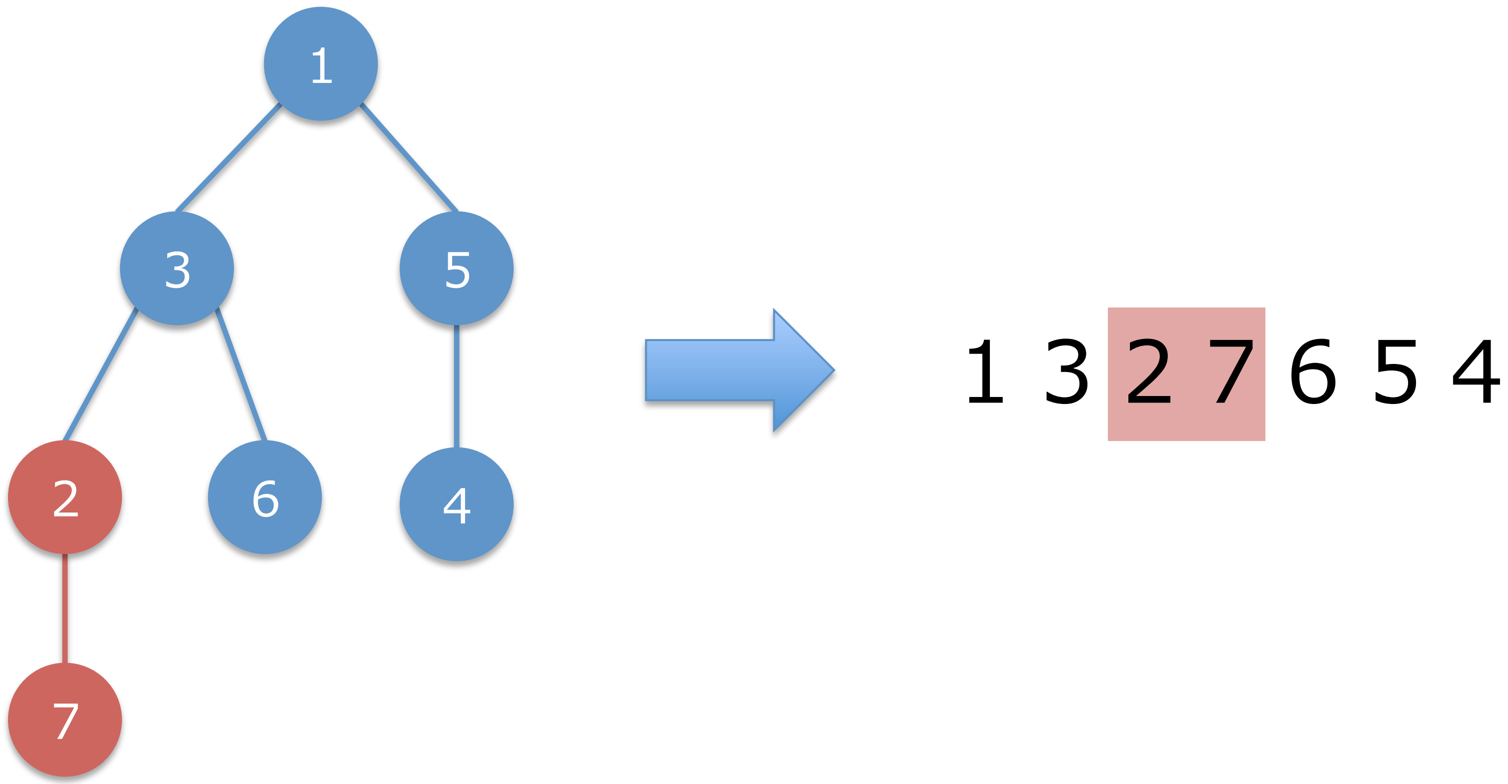
解法

- 行きがけ順の性質
  - 部分木が区間に対応している
  - さらに、区間の先頭は部分木の根



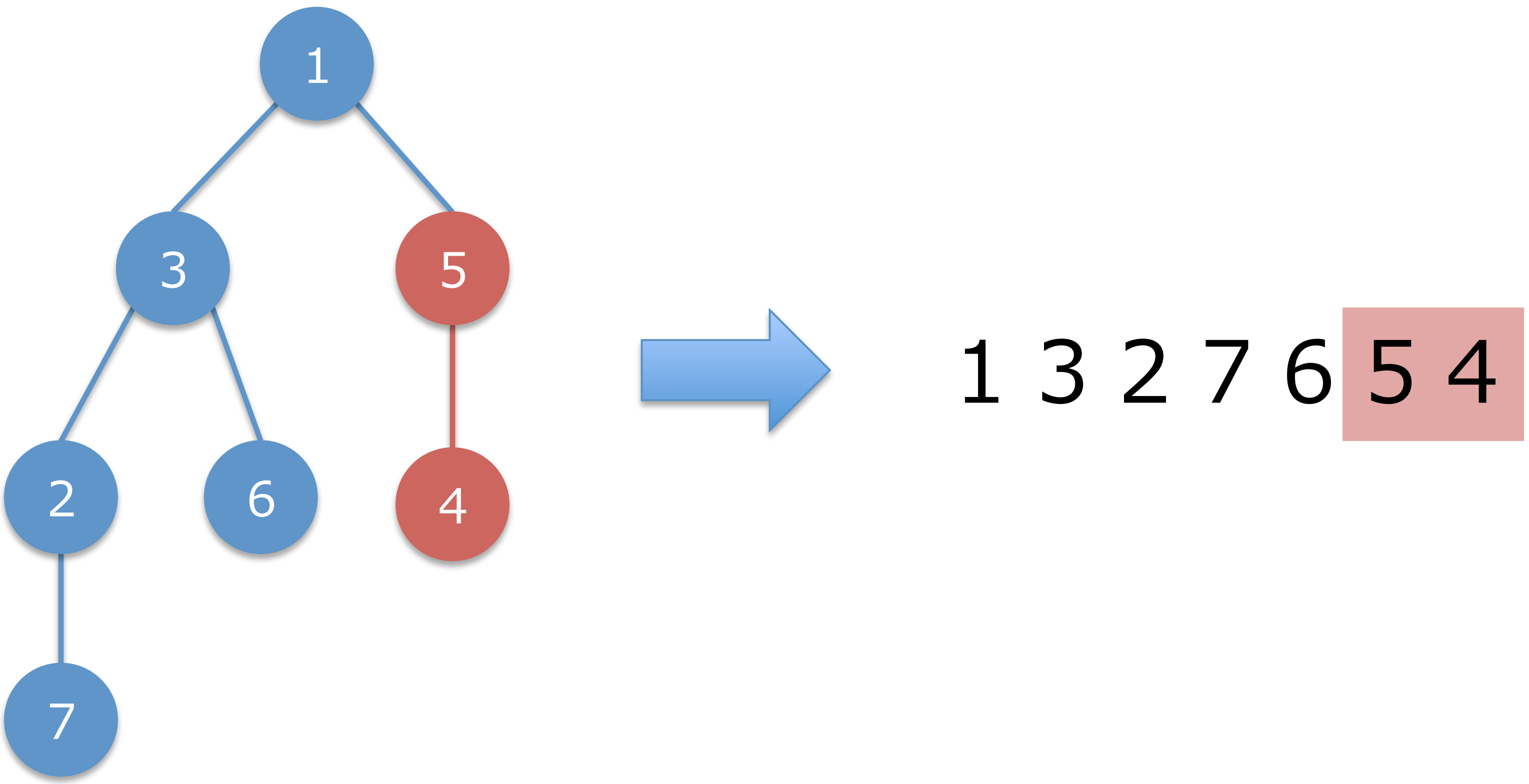
解法

- 行きがけ順の性質
  - 部分木が区間に対応している
  - さらに、区間の先頭は部分木の根



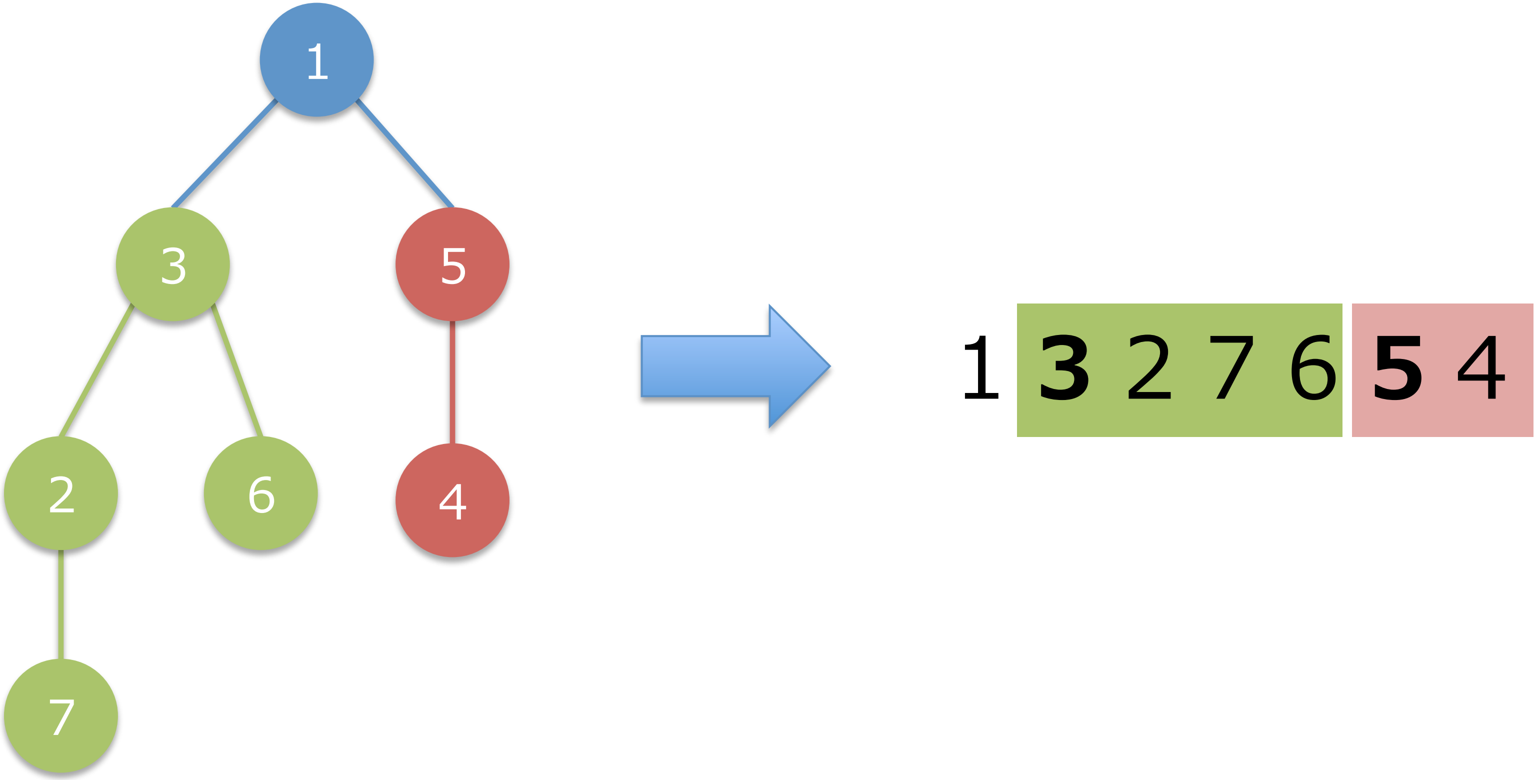
解法

- 行きがけ順の性質
  - 部分木が区間に対応している
  - さらに、区間の先頭は部分木の根



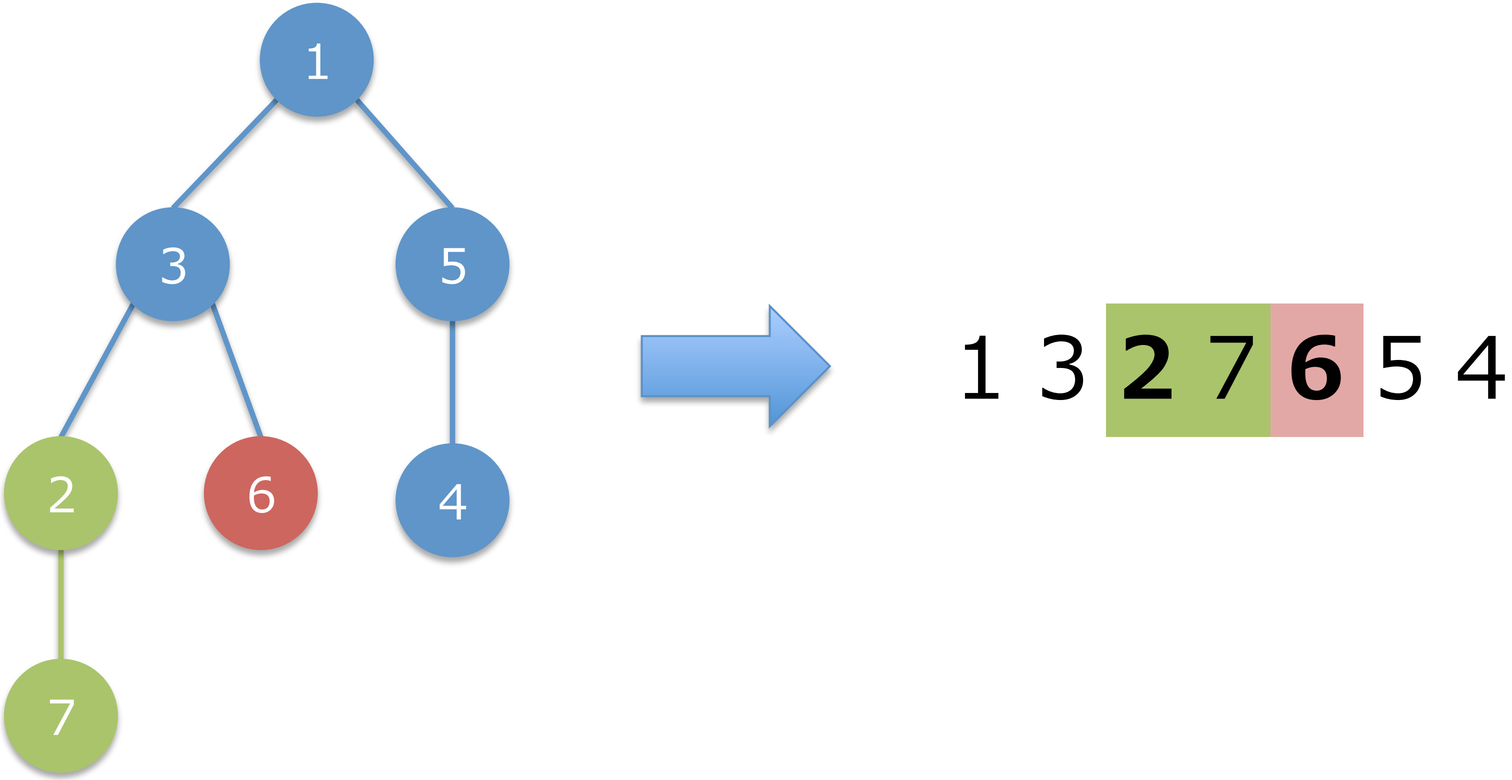
解法

- 辞書順最小の行きがけ順が満たすべき性質
  - 各頂点の子の区間は、先頭の小さい順に並んでいる



解法

- 辞書順最小の行きがけ順が満たすべき性質
  - 各頂点の子の区間は、先頭の小さい順に並んでいる



解法

- 区間DPを計算する
- 状態
  - DP_tree[L][R]
    - C_L～C_Rからなる木（C_Lが根となる）が何通りあるか
  - DP_forest[L][R]
    - C_L～C_Rからなる森（C_Lが最も左の木の根となる）が何通りあるか
- 遷移
  - DP_tree[L][R] = DP_forest[L+1][R]
    - 森に根をくっつける
  - DP_forest[L][R] = 以下のmin
    - min(DP_tree[L][i-1]+DP_forest[i][R])
      - $L < i \leqq R$ かつ $C_L < C_i$ を満たすような i について
      - 森の左に木をくっつける
    - DP_tree[L][R]
      - 木は森である

解法

- 区間DPを計算する
- 初期化
  - DP[i][i] = 1
- 計算量
  - $O(N^3)$

H問題 解説
「焼肉の達人」

CODE FESTIVAL 2015 本戦

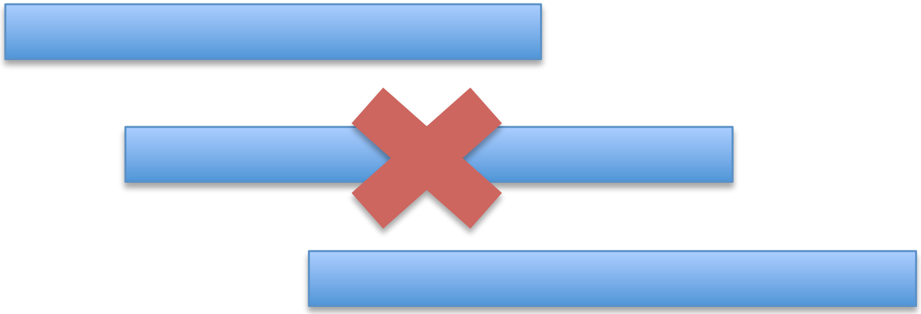
問題概要

- 区間がN個与えられる
- 区間はすべて[0,M]の区間に収まっている
- いくつかの区間を選んで「ちょうど１つの区間だけに覆われている部分」の長さの和を最大化せよ

- 1 ≦ N ≦ 10^5
- 1 ≦ M ≦ 10^5

解法

- 重要な考察
  - 区間が３つ以上重なっている部分ができるように選ぶのは無駄
  - そこに重なっている区間のうち、
    - 左端が最小のもの
    - 右端が最大のもの
  - の２つ（または１つ）以外は選ぶだけ損



解法

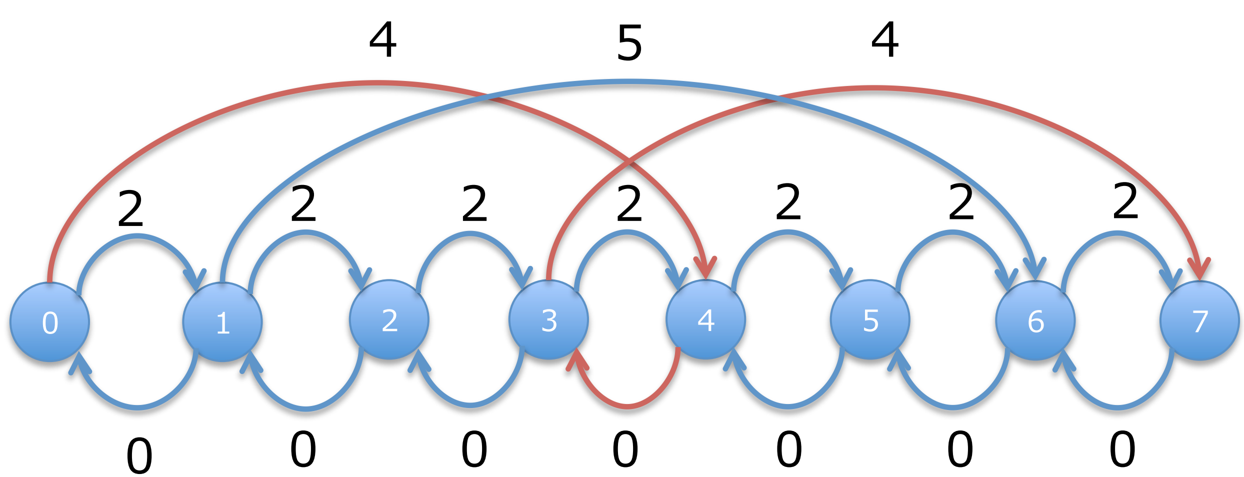
- 各部分に重なっている区間の個数は０か１か２
  - これが１になっている部分の長さの和を最大化したい
- 以下のようにペナルティを設定してみる
  - 重なっている区間の個数が０か２：ペナルティ２
  - 重なっている区間の個数が１：ペナルティ１
- このとき「2M - ペナルティの合計」が重なっている区間が１になっている部分の長さの和となる
- つまり、ペナルティを最小化する問題が解ければ良い

解法

- ペナルティの最小化は以下のような問題に帰着できる
  - 以下の２種類の区間からいくつかの区間を選んで[0,M]の区間を全て覆うとき、最小のコストは？
    - 元の問題で与えられた区間（コスト＝区間の長さ）
    - [x,x+1]の区間（コスト＝２）（合計M個）

解法

- アルゴリズム
  - 以下のように辺を張ったグラフで頂点0から頂点Mへの最短経路をダイクストラ法で求める
    - 頂点は各座標（0~M）
    - 各区間（前のスライドに書いたもの）について、左端→右端へコストが[区間の長さ]の辺を張る
    - 各xについて、x → x-1 へコストが 0 の辺を張る



I問題 解説
「風船ツリー」

CODE FESTIVAL 2015 本戦

問題概要

- 辺に長さのついた頂点数Nの根付き木が与えられる
- ある頂点の高さは根までの辺の長さの和
- 木の高さは頂点の高さの最大値
- Q個の整数が与えられる
- それぞれの整数（Xとする）について、木の高さをちょうどXにするために切る必要のある辺の本数の最小値を求めよ

- 制約
  - 2 ≦ N ≦ 10^5
  - 1 ≦ Q ≦ 10^5
  - 1 ≦ 辺の長さ ≦ 10^4

解法

- 辺を切ったときの木の高さとして考えられるものは高々N通りしかない（各頂点の高さ）
- 頂点vを木の中で最も高い頂点にするために切る必要のある辺の本数の最小値をそれぞれの頂点についてもとめればいい

解法

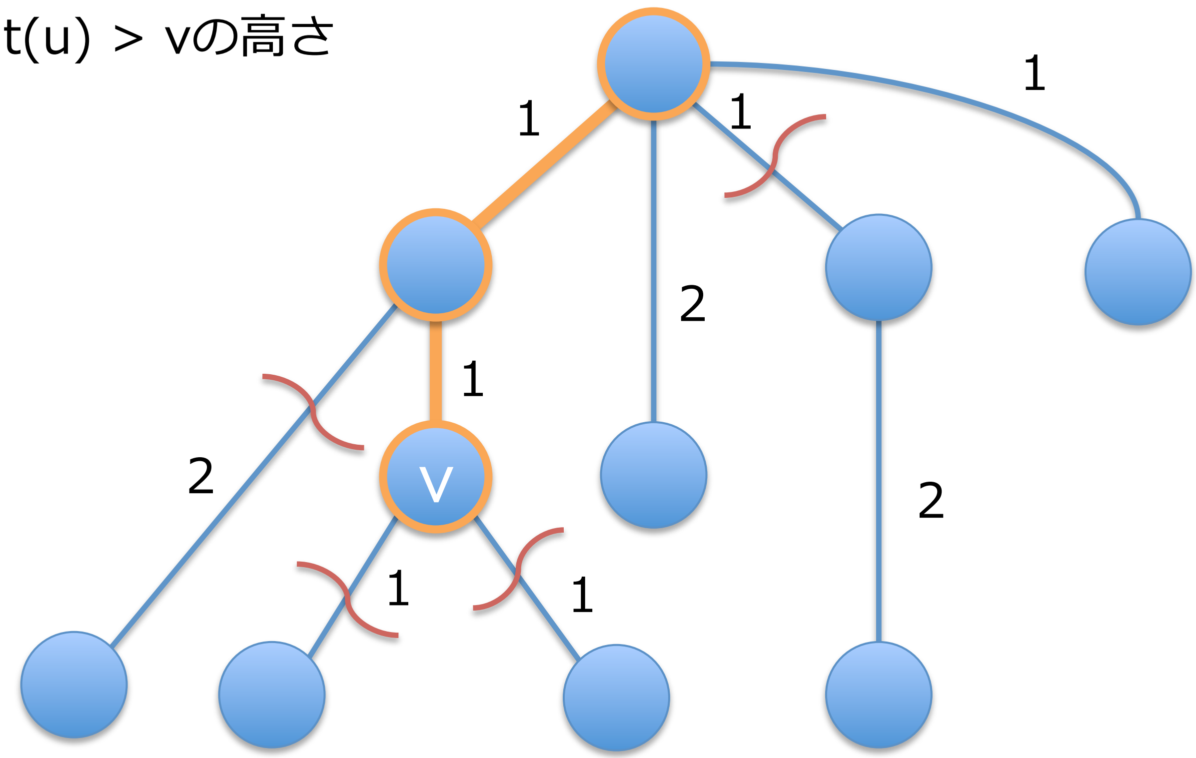
- まず各頂点の高さを求めておく
  - 根から順に計算していけばいい
- このとき、高さの大小関係しか辺を切る本数には影響しないため、圧縮をしておく
  - i番目に大きい高さの値をiに変換する
  - すると、高さが1~NのN種類になる
    - 同じ高さが複数ある場合もう少し少なくなることはある

解法

- さらに、各頂点vについて、子孫のうちの高さの最大値 max_height(v) も求めておく
  - 葉から順に計算していけばいい

解法

- 頂点vを最も高い頂点にするために切る辺は？
  - 根からvまでのパスに含まれる辺は切ってはいけない
  - 根からvまでのパスに含まれる頂点の子(uとする)のうち以下の条件を満たすものを切ると良い
    - vの先祖でない
    - max_height(u) > vの高さ



解法

- アルゴリズム
  - 先祖の直接の子のmax_heightの集合を更新しながら根からDFS
    - 頂点vに訪れた時は、この集合に含まれる数のうちvの高さより大きいものの個数を求めれば良い
  - 集合はBIT(Fenwick tree)で管理すれば良い。
    - 高さを1~Nの値に圧縮しておけば各更新が O(log N)
  - 具体的な更新方法
    - 頂点vに訪れたとき：全ての子のmax_heightを集合に追加
    - 頂点vから子cに潜るとき：cのmax_heightを集合から削除
    - 子cから頂点vに戻ってきたとき：cのmax_heightを集合に追加
    - 頂点vから親に戻るとき：全ての子のmax_heightを集合から削除
  - 計算量は全体で O(N log N) となる

J問題 解説
「N個のバケツ」

CODE FESTIVAL 2015 本戦

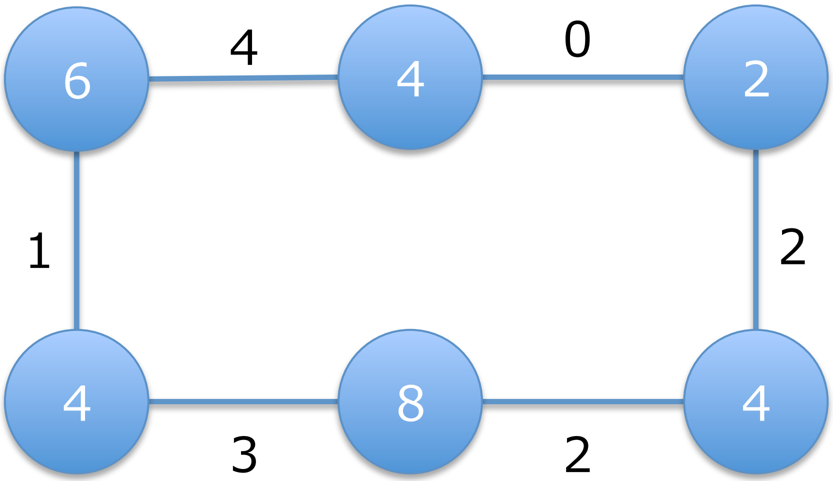
問題概要

- N人のチームメンバーが円周上に並ぶ
- 隣り合った人の間に水を入れたバケツを置く
- 各バケツに入れる水の量は自由に決められる
- バケツを全て持ち上げられれば水の総量がスコアになる
- 各メンバーは、自分の隣にある２つのバケツの水の量の平均が自分の力の値よりも小さければ持ち上げられる
- メンバーが1人変更になるというクエリが Q 個ある
- 各クエリごとに、メンバー変更後のチームで、水の量をうまく選んだ時のスコアの最大値を求めよ

- 制約
  - 3 ≦ N ≦ 10^5、N は偶数（追加点制約ではこの制約はない）
  - 1 ≦ Q ≦ 10^5
  - 1 ≦ メンバー i の力 ≦ 10^4

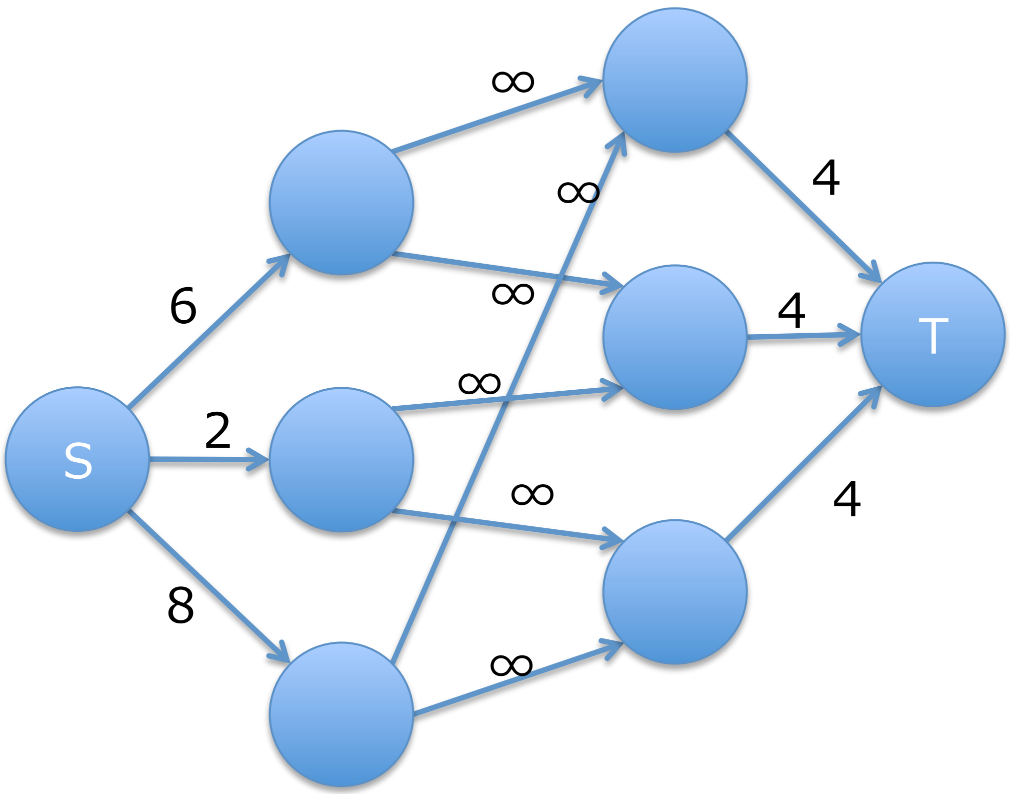
解法

- グラフ問題に帰着してみる
  - 頂点はメンバー、辺はバケツ
  - 辺に 0 以上のコストを設定する
  - 頂点につながっている 2 つの辺のコストの和が頂点に定められた制限（力の値× 2 ）を超えないようにしたい
  - 辺のコストの和の最大値は？



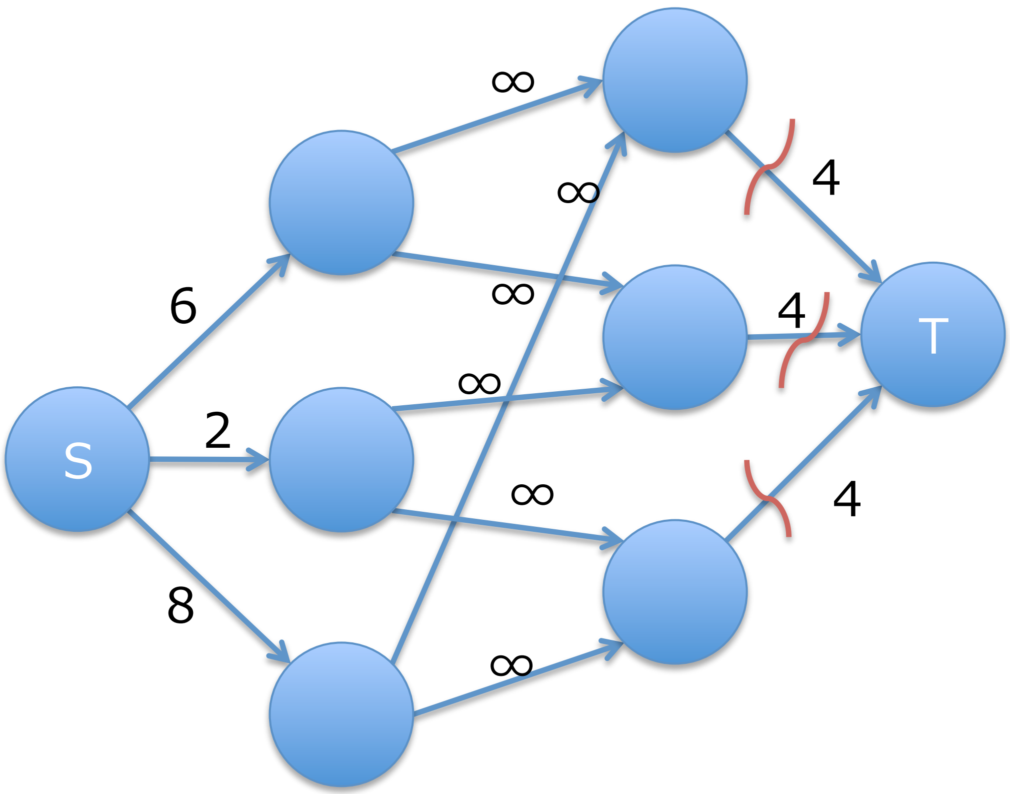
解法

- 最大流問題に帰着してみる
  - メンバー番号の偶奇で左右に分け、ソース/シンクから各メンバーへの辺と、隣り合ったメンバーの間の辺を張る
  - 真ん中の辺の流量が各バケツに入れる水の量に対応する



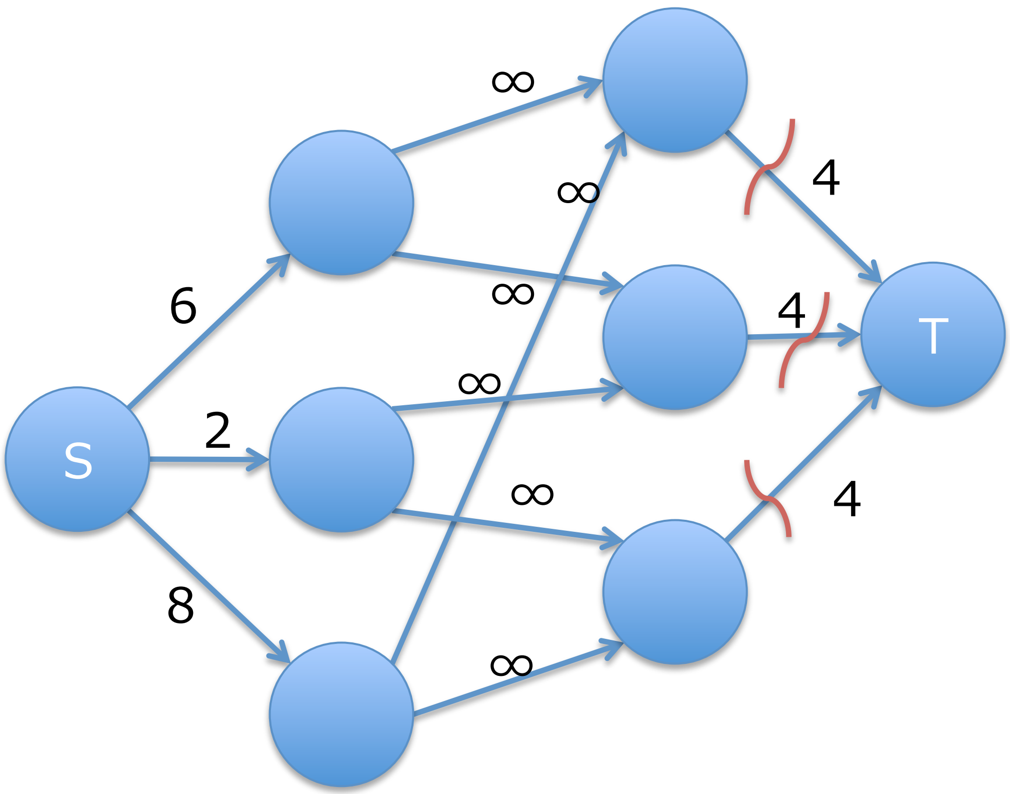
解法

- 最小カット問題に帰着してみる
  - 最大フロー最小カット定理より、最大流＝最小カット



解法

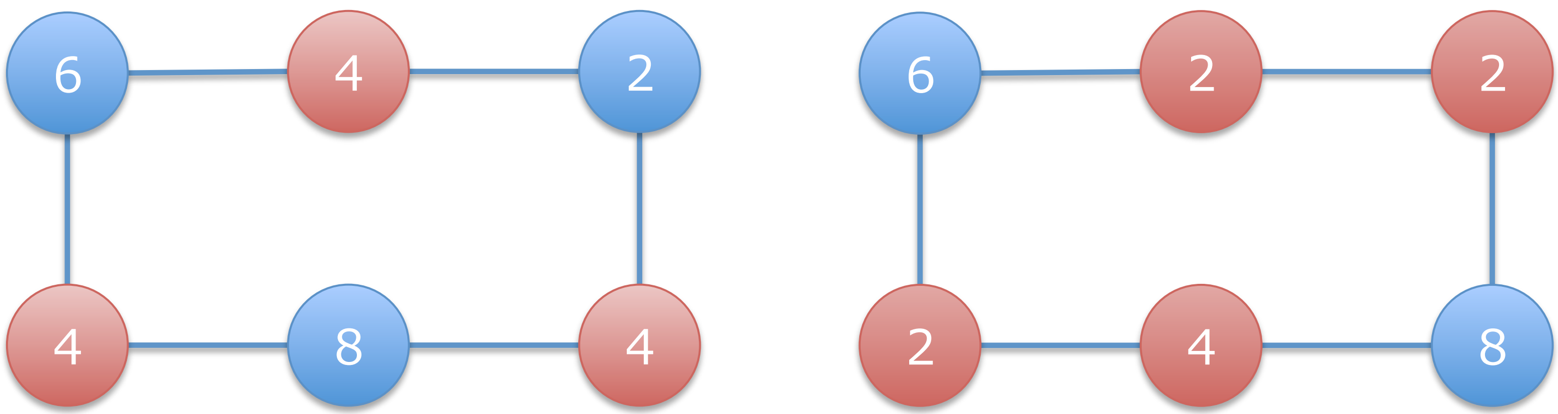
- 最小カットの構造を考察してみる
  – 各バケツごとに、２人のメンバーのうち少なくとも片方の辺を切らなければならない（そうしないとS-Tパスができてしまう）



解法

- つまり？
  - 隣り合った２人のメンバーのうち少なくともどちらか一方が選ばれるように何人かのメンバーを選ぶことを考える
  - このとき選ばれるメンバーの力の和の**最小値**がスコアの最大値

メンバーの力の和が最小となるような選び方の例

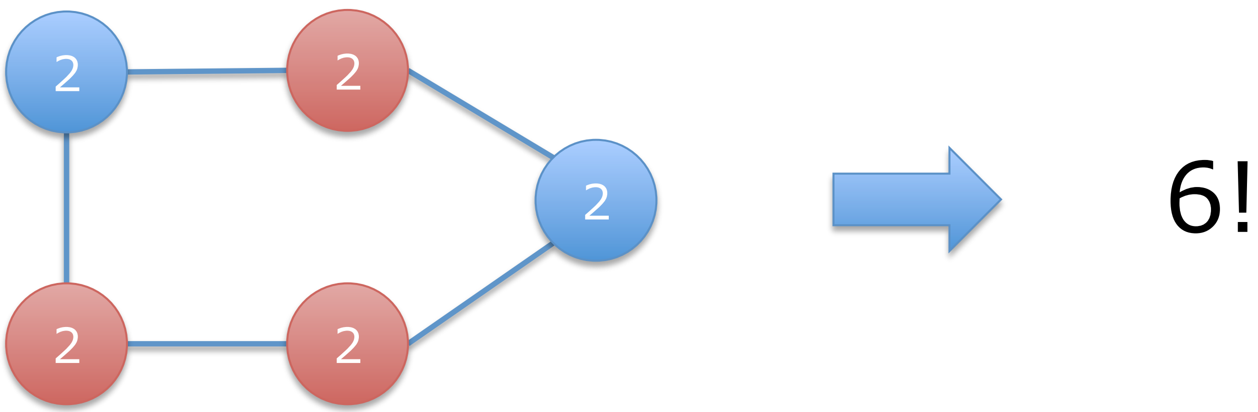


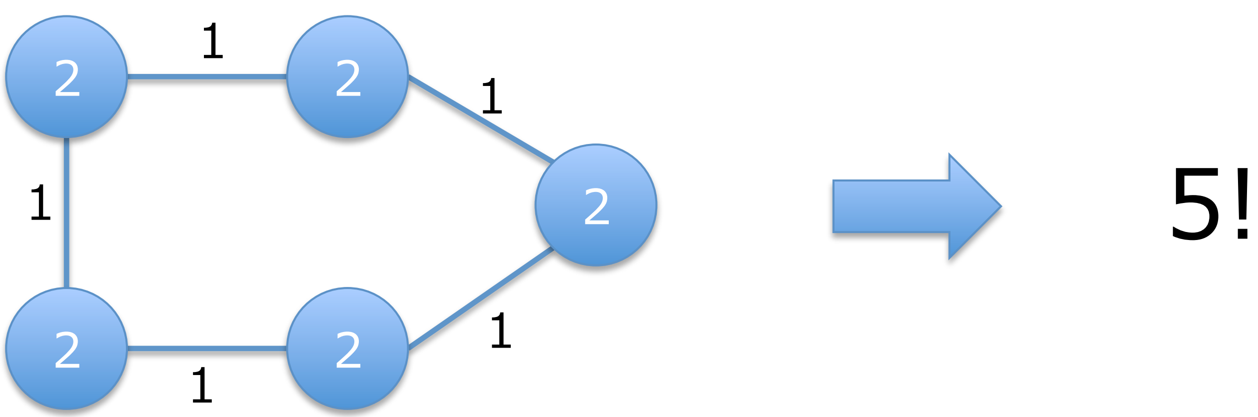
解法

- 以下のデータを計算するsegtreeを用いる
  - ある区間のメンバーの選び方についての以下の４通りの値
    - 左端も右端も選ぶときの力の和の最小値
    - 左端を選び、右端を選ばないときの力の和の最小値
    - 左端を選ばず、右端を選ぶときの力の和の最小値
    - 左端も右端も選ばないときの力の和の最小値
- 1~N の区間についてのデータの、左端も右端も選ばないとき以外の値のうちの最小値が答えとなる
- メンバー更新・答えの計算がともに O(log N) でできる
- 全体の計算量は O(N log N + Q log N) となり、満点を得ることができる

追加点解法

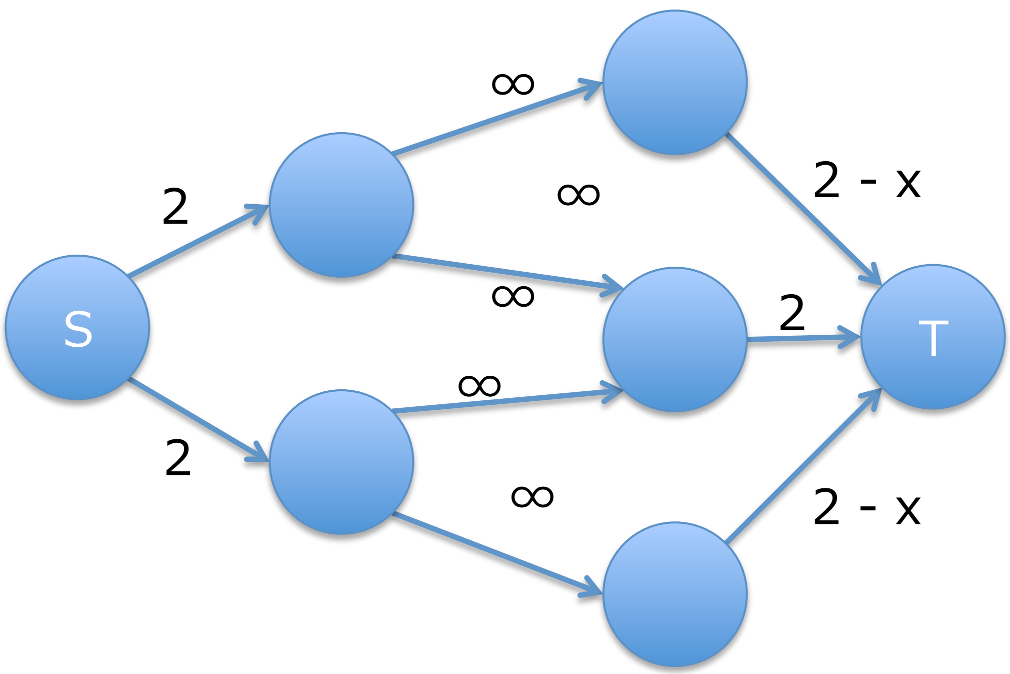
- Nが奇数の場合、先ほどの解法だとダメ





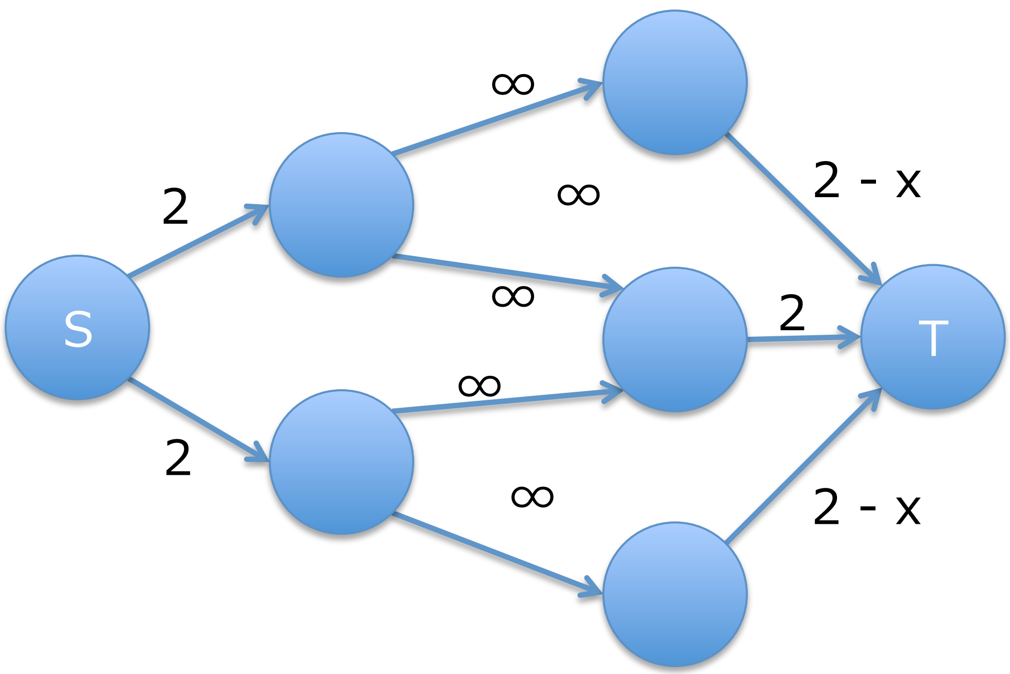
追加点解法

- そもそも最大流のグラフがうまく構築できない
- １ヶ所辺を切って、直線グラフにすると構築できる
  - 切った辺のコストを x とする



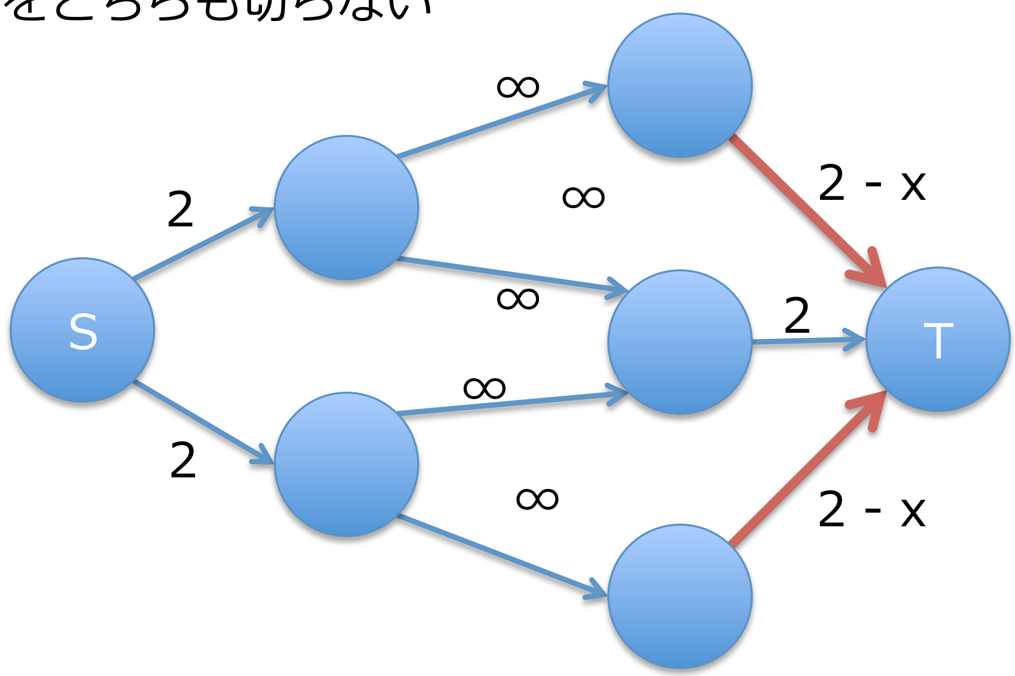
追加点解法

- x を変動させたときの、「mincut+x」の最大値が答え
  - 考えうる x (0~min(両端のメンバーの力)) について全て計算すれば正しい答えは求まりますが、これだとTLE



追加点解法

- mincutの形に注目
  - x を変動させたとき、mincutで切る辺の集合としてありうるものは 3 通りしかない
    - 両端（図の赤い辺）をどちらも切る
    - 両端のうち片方を切る
    - 両端をどちらも切らない



追加点解法

- パターンが切り替わるところの x の値を求める
  - それぞれのパターンの式は以下の通り
    - 両端をどちらも切る：定数 - 2x
    - 両端のうち片方を切る：定数 - x
    - 両端をどちらも切らない：定数
  - 全て x に関しての一次式となる
  - これらのうちの最小値が各 x でのmincutとなる
  - これらのうちの２つの一次式の交点 x 座標（3 通り）と x の上限・下限についてそれぞれmincut+xの値を求め、それらの最大値を求める
- 各クエリごとの計算量を O(log N) に抑えられた
- これで追加点を得ることができる