ニコニコ文字列判定

問題概要

- 4 文字の文字列 s が与えられる
- ある数字 x, y が存在して s は xyxy か？

解法

- 1 文字目と 3 文字目、2 文字目と 4 文字目が同じ文字かどうかを調べる

2525文字列分解

問題概要

- 文字列 s が与えられる
- s をいくつかの部分列に分解して、
  それぞれの要素が 2525 文字列にせよ
  - 2525文字列： “25”, “2525”, “252525”, …
- 最小でいくつに分解すればいいか？
  - 不可能なら -1 を出力
- 1 ≦ |s| ≦ 2,525

考察

- 空文字列も 2525文字列 ということにする
  - そのような分解が最適ということはありえない

- 25…25 の状態 (A) と 25…2 の状態(B) に列を分類する
  - A は “2” を追加して B に遷移
  - B は “5” を追加して A に遷移

解法

- 答え K を全探索
  - はじめ状態 A の列が K 個ある
  - s の先頭から 1 文字ずつ追加

- 以下の状態になったら失敗
  - 状態 A, B の列が足りない(遷移不可能)
  - 最終的に状態 B の列が残る

- O(|s|) でシミュレーション可能
  - 答えを全探索しても $O(|s|^2)$

おまけ

- 実はこの問題は「正しい括弧列を、どの括弧列の深さも最大 1 となるように分解せよ」と言い換えられる
  - “2“ は “(” と，”5” は “)” と対応
- 各深さごとに 1 つの括弧列に分解すればよく、以下の 2 つを調べれば十分
  - 正しい括弧列かどうか
  - 深さの最大値はいくつか
- O(|s|) で実行可能

Kill/Death

問題概要

- 長さ N の数列 killA = [0, 0, ..., 0] と deathA = [0, 0, ..., 0]
長さ M の数列 killB = [0, 0, ..., 0] と deathB = [0, 0, ..., 0]
がある
- 以下の操作を何回か行う
  - killA[i] と deathB[j] を同時に 1 加算する
  - killB[i] と deathA[j] を同時に 1 加算する
- 結果としてある killA と killB が得られた
- このときにありえる deathA, deathB の組み合わせは何通り？

考察

- ありえる deathA の個数と deathB の個数をそれぞれ求めて掛け合わせればよい
- killA[i] = killA[j] (i < j) ならば deathA[i] ≧ deathA[j] でなければならない
- それ以外のときは制約がない
- 自チームのデス数の総和は相手チームのキル数の総和
  - 制約に違反しないように deathA に割り当てていけばよい

解法

- キル数が同じプレイヤーをグループ化する
  - k[i] := i 番目にキル数が多いグループの人数
- 動的計画法
  - dp[i][j] := [1, i) 番目のグループに j デス割り当てるときの組み合わせ数
  - dp[i][j] -> dp[i+1][j+x] への遷移が（i 番目のグループに x デス割り当てる組み合わせ数）通り存在する

解法

- n 人のグループに x デス割り当てる組み合わせの数
  - = 長さ n の非減少数列 A (A[i] ≧ 0) で、総和が x になるものの個数
  - = x の**分割数**とは厳密に言えば違うけど気持ち的には同じもの
- 分割数は動的計画法で求まる
  - 長さ数列 [1] から初めて、以下の操作を繰り返す
    - 操作1: 全ての要素に +1 する
    - 操作2: 数列の末尾に 1 を付け加える
  - 補題: この操作列は非減少数列と 1:1 に対応する
    - この操作から動的計画法を構成できる
  - dp[n][m] := 長さ n の非減少列 A (A[i] > 0) で総和が m のものの個数
    - dp[n][m] -> dp[n][m+n] (操作1に対応する遷移)
    - dp[n][m] -> dp[n+1][m+1] (操作2に対応する遷移)
  - O(nm)
    - この問題では前計算しておけばいいので、全体の計算量には影響を与えない

計算量

- $O(NK_B^2 + MK_A^2)$
  - K_A, K_B は各チームの合計キル数 ($\leqq 1{,}000$)

ディスクの節約

概要

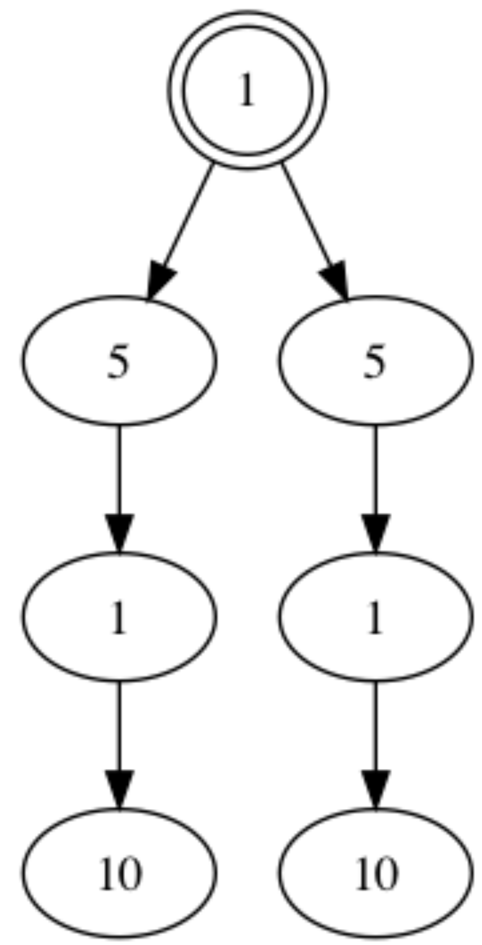
- 依存関係のある N 個のタスクが与えられる
- 依存先のタスクの実行結果がディスクに書き込まれていないとそのタスクは実行できない
- 依存関係は有向木になっている
- 実行結果はいらなくなったら消していい
- 根タスクを実行するまでに必要なディスクの空き容量は？

解法

- あるタスクを実行完了したとき、そのタスクに必要だった依存タスクの結果は消去してよい
  - あるタスクを2回以上実行する必要もない
- dp[S] := 実行済みのタスク集合が S の状態に至るまでの最大瞬間ディスク消費量
  - あるいは dp[S] := 結果がディスクに乗っているタスク集合が S の状態に至るまでの最大瞬間ディスク消費量 としてもよい
- $O(2^N N)$

想定誤答

- 「ある部分木を完全に処理してから別の部分木を処理する」という方針は誤り
  - 右：撃墜ケース
    - Expected: 12 (10 + 1 + 1)
    - Actual: 16 (10 + 5 + 1)



別解

- 解に対する二分探索ができる
  - 想定解法のDPと同様だが、その状態への到達可能性だけを持つだけでよい
- $O(2^N N\log(N))$
  - 通るが、実装によっては定数倍の高速化が必要かも

ニワンゴくんの家探し

問題概要

- N 頂点の木が与えられる
  - 2 ≦ N ≦ 2,000
  - どの頂点も次数は 5 以下
- どこかにニワンゴくんの家がある
- 2 頂点を指定すると、家からの位置関係がわかる
- 14 回の質問で家の位置を特定せよ

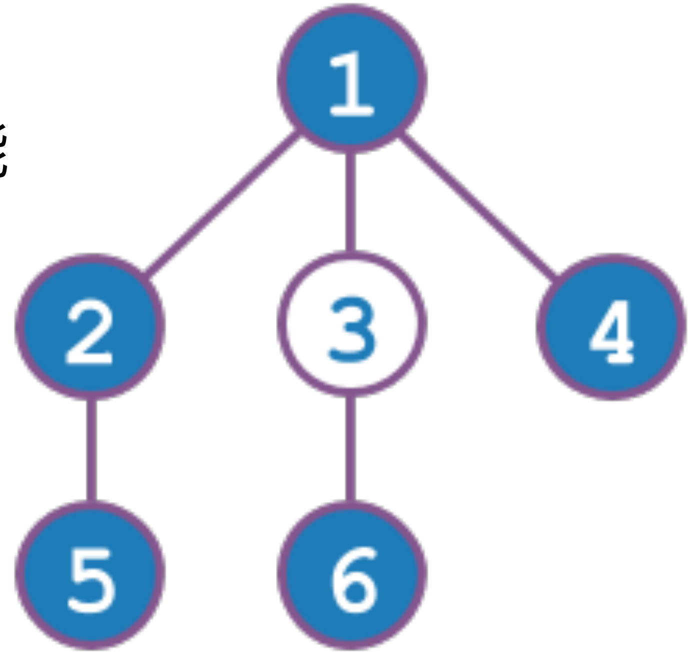
考察

- N = 1 ならば自明
- N = 2 ならば、2 点について質問すればいい
  - 帰ってきた番号が答え
- 以降 N > 2 を仮定

- 頂点 u, v について質問すると、u, v をつなぐパスの真ん中を境としてN 個の頂点を分類可能
  - 真ん中は辺、あるいは頂点のどちらかとなる
  - とりあえず頂点が真ん中になるようにしておく

考察

- 質問により 3 種類の頂点に分類可能
  - ある頂点 X を境として u 側
  - ある頂点 X を境として v 側
  - u, v から等距離
- X に隣接した 2 点を u, v として選べば必ず X を頂点として指定可能
  - X が葉である場合は成立しないが そのような質問はあまり有効でない
  - どのような X を選ぶべきか？



方針

- 頂点 X が部分木 a, b, c, d, e を持つとする
  - $|a| + |b| + |c| + |d| + |e| + 1 = N$
  - $N > |a| \geqq |b| \geqq |c| \geqq |d| \geqq |e| \geqq 0$
  - a, b, c, d, e について、 X に隣接する頂点を根と定義する

- これから以下を示す
  - うまく質問すると基本的には以下が可能
    - 1 回の質問で候補の数を N/2 以下に
    - あるいは、2 回の質問で候補の数を N/4 以下に

方針

- X として重心を選ぶことにする
  - $|a| + |b| + |c| + |d| + |e| + 1 = N$
  - $N/2 \geqq |a| \geqq |b| \geqq |c| \geqq |d| \geqq |e| \geqq 0$
- $|a| + |b| \geqq N/2$ のとき
  - a, b の根について質問： 候補が $\max(|a|, |b|, N - |a| - |b|)$ 個以下に
  - $N/2 \geqq |a| \geqq |b|$, $|a|+|b| \geqq N/2$ より
    1 回の質問で候補を N/2 以下にできる

方針

- |a| + |b| < N/2 かつ |a| + |b| + |c| ≧ 3N/5のとき
  - a, b の根について質問：候補はmax(|a|, |b|, N - |a| - |b|)個以下に
  - a, b 側が選ばれれば 1 回の質問で候補 N/2 以下に

  - c, d が選ばれたとき，c, d の根についてさらに質問をすると max(|c|, |d|, |e| + 1) 個まで候補を減らせる
    - このとき，|c| < N/4 を満たすので c, d 側が選ばれれば 2 回の質問で候補を N/4 以下に
    - |d| + |e| + 1 ≦ 2N/5 より，|e| + 1 ≦ N/5 + 1/2 となり 10 ≦ N の場合、悪くても 2 回の質問で候補を N/4 以下にできる
      - N < 10 の成り立たない場合も候補の数は 2 まで十分小さくなる

方針

- $|a| + |b| < N/2$ かつ $|a| + |b| + |c| < 3N/5$のとき
  - a, b の根について質問：候補は$\max(|a|,|b|,N-|a|-|b|)$個以下に
  - a, b 側が選ばれれば 1 回の質問で候補を N/2 以下に

  - c, d の根についてさらに質問： $\max(|c|, |d|, |e| + 1)$ 個以下に
    - このとき，$|c| < N/5$ を満たすので c, d が選ばれれば 2 回の質問で候補を N/4 以下に
    - $|c| \geqq |d| \geqq |e|$ より，$|e| + 1 < N/5 + 1$ となり $20 \leqq N$ の場合、悪くても 2 回の質問で候補を N/4 以下にできる
      - $20 < N$ の成り立たない場合も候補の数は 4 まで小さくなる

方針まとめ

- 戦略は以下の通り
  - a, b の根について質問
    - 候補の数が N/2 以下になれば残った部分木に対して同じ問題を解く
    - そうでなければ c, d の根について質問し、残った部分木に対して同じ問題を解く

- N ≧ 20 なら以下が成立
  - 1 回の質問で候補の数を N/2 に、あるいは
    さらに質問をすればサイズを N/4 以下にできる
- N < 20 でも
  - 1 回質問すれば候補の数を N/2 以下に、あるいは
    さらに質問をすれば候補の数を 4 以下にできる
  - 4 以下なら 2 回の質問で必ず特定可能
- N ≦ 2000 より、14 回あれば十分余裕がある

解法

- 前述の戦略を実装する
  - 実際には毎回重心を探し、重心の直接の子のうち
    サイズが大きい順に2 つ選んで質問を繰り返してもよい
  - 7 回の質問で残り 20 個未満にできる
  - ここから 5 回も質問をすれば必ず特定可能

- 必ず 12 回以下の質問でうまくいく
- 計算量は O(N)
  - N(1 + 1/2 + 1/4 + …) ≦ 2N