Harvest 解説

問題概要

- 円周上に人がいる
- りんごの木も植えてある
- 人間が等速度で時計回りに動いていく
- りんごの木は C 秒経つと実をつける
- ある時刻までにある人間が取ったりんごの個数は？というクエリに答えていく

問題概要

- 円周上に人がいる
- りんごの木も植えてある
- りんごが等速度で反時計回りに動いていく
- りんごの木は C 秒経つと実をつける
- ある時刻までにある人間が取ったりんごの個数は？というクエリに答えていく

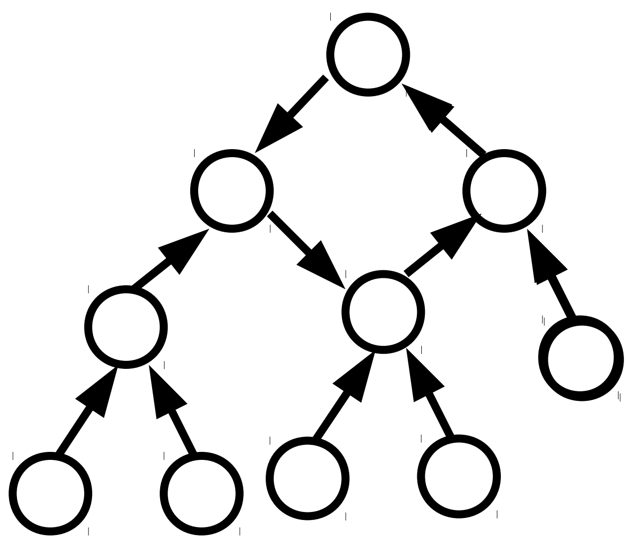
subtask 1

- C 秒経つと新しい実をつける

  →人間 x にりんごを渡した木が次にりんごを渡す人間は･･･
  →x から C 以上反時計回りに進んだあと初めて出てくる人間
- x の次に y に渡す, というときに, x->y の辺を貼ったグラフを考える

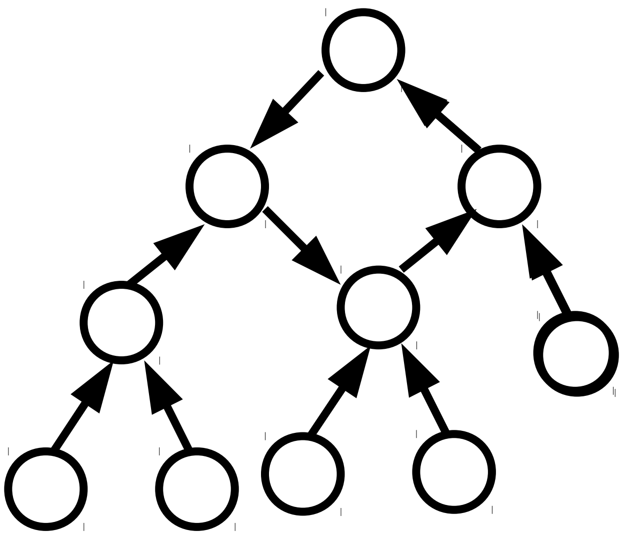
subtask 1

- 各頂点の出次数が 1 のグラフ
→ サイクルと木が集まってできたグラフ



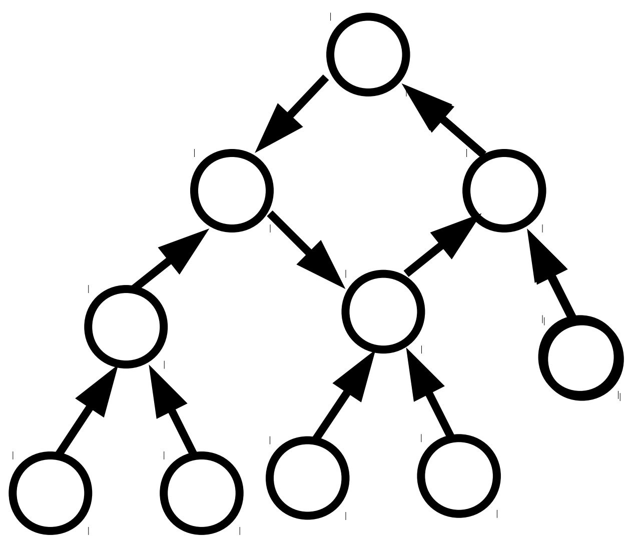
subtask 1

- 各りんごの木は, 適当な頂点からスタートし, どんどんりんごを配っていく



subtask 1

- 木の部分の頂点には, 一度しか配らない
- サイクルの部分の頂点には, 一定周期で配る



subtask 1

- 木の部分の頂点に関するクエリ
どのりんごが時刻いくつで届けられるか管理しておけばいい
- サイクルの部分の頂点に関するクエリ
どのりんごが時刻いくつで最初に届けられるか管理しておけばいい

- O(NM+QM) とか, なんでもいい

subtask 2

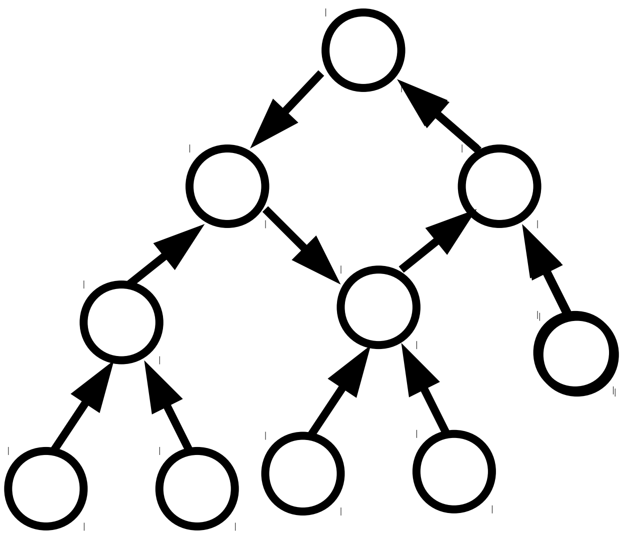
- 時刻がでかい
- どのりんごも到達できる頂点には一度以上到達してる

subtask 2

- 木の部分の頂点に関するクエリ
  その頂点に到達しうるりんごの個数を数えればいい
- それは, 木においてその頂点とその子孫からスタートするりんごの数が数えられればいい
- 最初に一回dfsするだけ

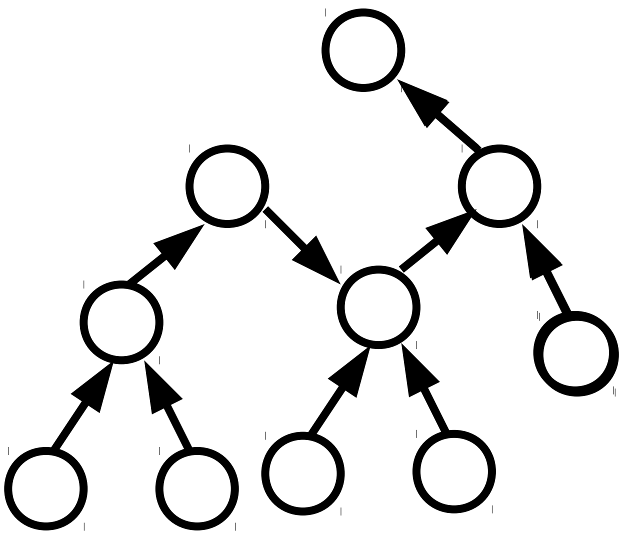
subtask 2

- サイクルの部分に関するクエリ
- サイクルを適当な辺で切って，木にしておく



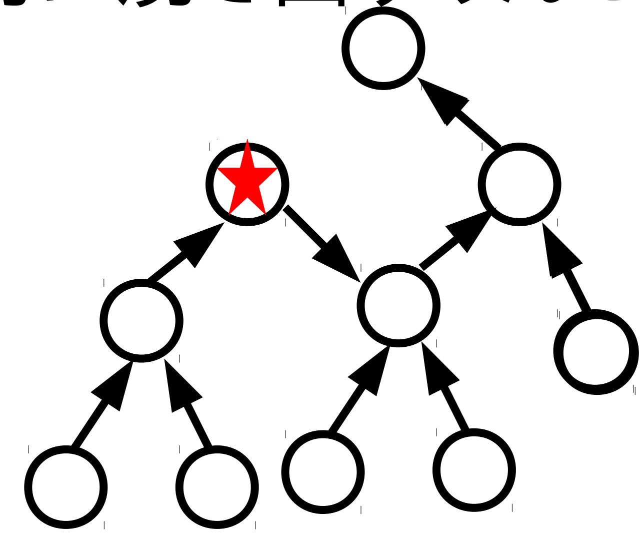
subtask 2

- サイクルの部分に関するクエリ
- サイクルを適当な辺で切って，木にしておく



subtask 2

- そうすると木の問題になる
- 本来サイクルをループするりんごは, 切ったところから定期的に湧き出すりんごとして処理する.



subtask 2

- 定期的に湧き出すりんごを何個獲得するかが分かればいい
- 湧き出してくる頂点を x とおく
- 頂点 Y で時刻 T までに得られるりんごの数

  →頂点 X で時刻 T-(xからyへ至るまでの時間) までに得られるりんごの個数

subtask 2

- 結局, 頂点 x に関するクエリだけ答えられればいい
- 頂点 x からあるりんご r が湧き出すタイミングは, a_r + Dt (t=0,1,2,…) という形なる (D はサイクルを一周するのにかかる時間で, r に寄らない定数)
- 時刻 T のときのクエリの答えは ∑(T-a_r)/D+1 となる(割り算は切り捨て)

subtask 2

- a_r=b_r*D+c_r
- T=p*D+q とおく
- c_r<=q の場合 (T-a_r)/D+1=p-b_r+1
- c_r>q の場合 (T-a_r)/D+1=p-b_r
- よって, a_r を c_r の昇順にソートしておき, 適切に累積和を計算すれば, 一つの T に対しては O(logM) で答えが求められる.
- 計算量は O((N+M)log(N)+Q(logM)) とか

subtask 3

- 木の頂点に関するクエリ
- その木の根を x とする, クエリで聞かれた頂点を y とする.
- 時刻 T までに y に到達するりんごの個数=時刻 T+(yからxに至るまでの時間)までに x に到達する, y に到達可能なりんごの個数

subtask 3

- 木の頂点にオイラーツアーで順番をつけておく
- y に到達可能なりんご→始点となる頂点がオイラーツアー順である範囲に収まるりんご
- 各りんごに対し, (始点となる頂点のオイラーツアー順序)と, (x に到達する時刻) をプロットすると, 二次元平面の矩形内の点の個数をカウントする問題になる

subtask 3

- クエリがオフラインなので, 平面走査+BIT でできる
- O((M+Q)logM)

subtask 3

- サイクルの頂点に関するクエリ
- 先ほどと同様, 定期的に湧き出すりんごを処理できればいい
- 求めたいものは, $\sum \max((T-a_r)/D+1,0)$

subtask 3

- sub2 とほぼ同様にできる.
- 各りんごを, a_r の値と a_r%D の値でプロットすると, 二次元平面の矩形領域にある点の重みの総和を数えることになる.
- 木の頂点と同様に, 平面走査+BIT で O((M+Q)logM) でできる

subtask 3

- 全部合わせて, $O((N+M)\log N+(M+Q)\log M)$ でできる.