



Mígratíon Plan

Edítoríal : blackyukí

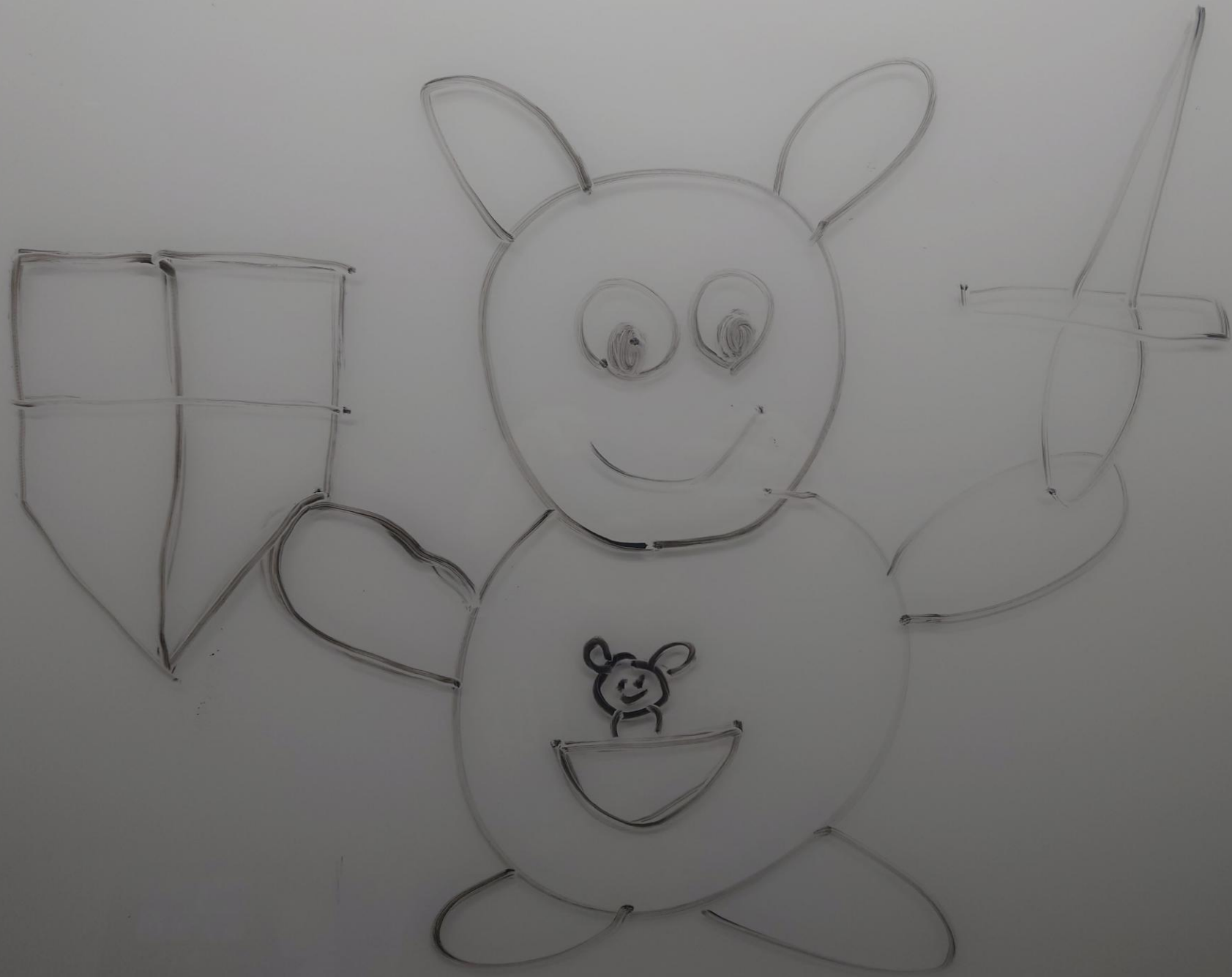
問題概要

- 根付き木が与えられる
- 3種類のクエリを処理
 - 深さ X の頂点に住んでいるビーバー全員が、今いる頂点の祖先である深さ Y の頂点に移動
 - 頂点 X に住むビーバーの数が増える
 - 頂点 X に住むビーバーの数を答える



勇者ビ太郎

絵: yuto1115



小課題 2

$N \leq 20$

小課題2

- N が小さい
- 愚直シミュレーション
- クエリ2,3 は $O(1)$ で処理できる
- r 回登った祖先を前計算でメモしておく、クエリ1は $O(N)$ で処理できる

小課題 1

$D=1$

小課題1

- クエリ1で $O(N)$ かけていると間に合わない
- ビーバーが住んでいない町を地下に埋める
- クエリ1で1つの町を処理する度にその町は消滅
- 地下に埋まった町が復活するのは、
クエリあたり高々1回

→全体 $O(N+Q)$ に抑えられる

小課題 3

$D \leq 20$

小課題3

- ・ 小課題2の実装を、小課題1のアイデアで高速化
- ・ ビーバーが住んでいない町を地下に埋めると
全体計算量が $O((N+Q)D)$ に抑えられる

小課題3

…なぜか？

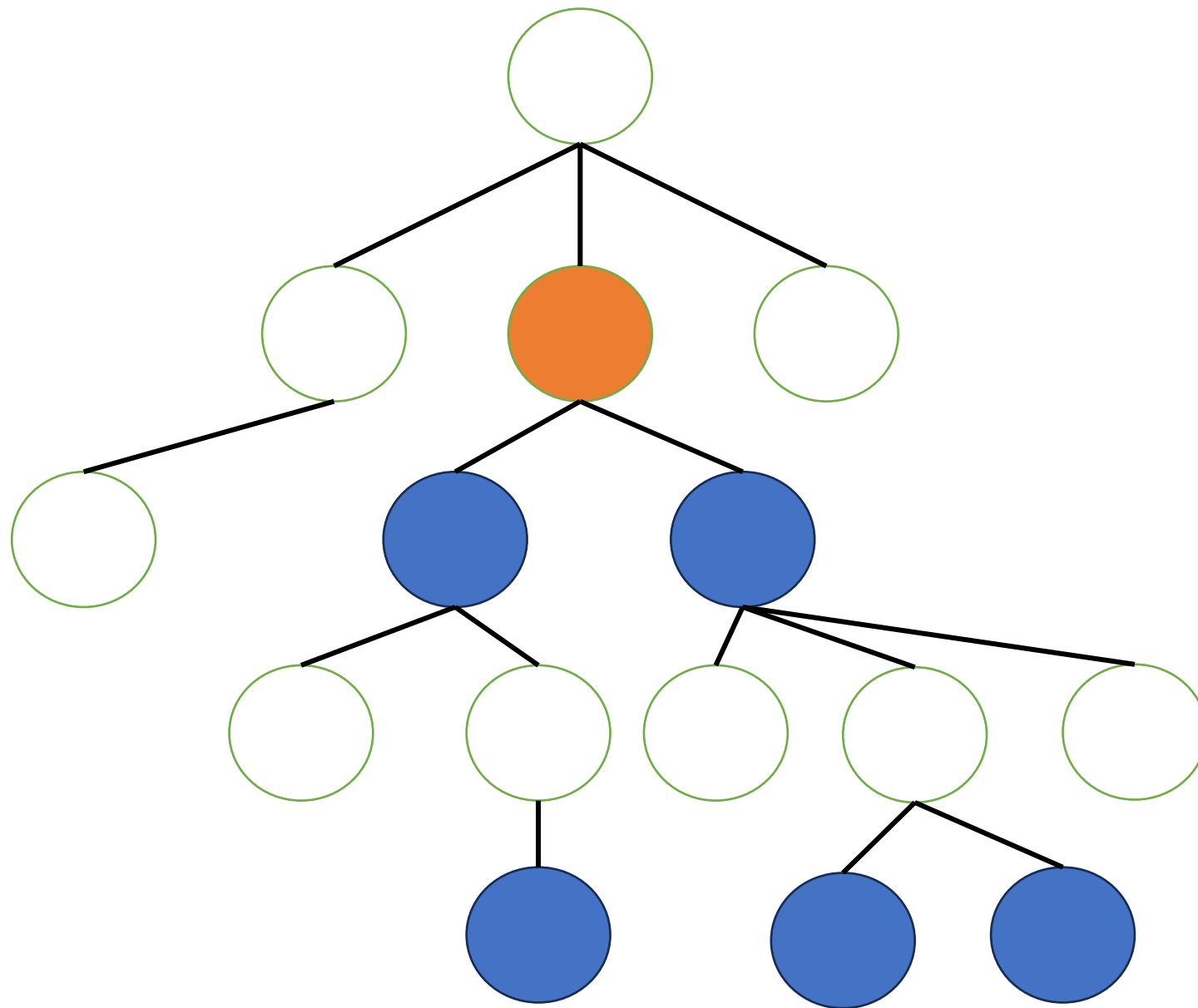
- 1匹のビーバーが移動できる回数は高々 D 回
- クエリ1で1つの町を処理する度にその町は消滅
- 地下に埋まった町が復活するのは、
全部で高々 $((N + Q)D)$ 回 ← 高々 $(N+Q)$ 種類のビーバー
→ 全体 $O((N+Q)D)$

小課題 4

加算クエリがない
&
回答クエリが 5 個以下

小課題 4

- これまではクエリ 1 で時間を使って、クエリ 2,3 は楽をする方針だった
- 小課題 4 は逆にクエリ 1 を高速に処理してクエリ 3 は時間を使っていいよ、と言ってそう
- 現在町 X に住んでいるビーバーがどこ由来か？
ということを考える



小課題 4

- 結局、重要な情報は、
「元々深さ X にいたビーバーの現在の深さ」
- 具体的な町の番号は関係ない
- 深さ X の町、という抽象的な頂点を作り、
クエリ 1 が飛んでくる度にマージすれば良い

→ 計算量は $O(D \log D + Q_3 N + Q)$

小課題 5

回答クエリが 5 個以下

小課題 5

- 深さ X の抽象的な頂点に加え、
「時刻 T に追加された深さ X の頂点」というものを考える
- 同様にマージしていくことができる

小課題 6

加算クエリがない

小課題 6

- 回答クエリも高速に処理する必要がある
- 「頂点 x の部分木の中で、
深さが $d_1, d_2, \dots$ である頂点の総和」

小課題 6

- 頂点 x の部分木 $\rightarrow$ オイラーツアーで区間
- 深さが $d_1, d_2, \dots$
 - $\rightarrow$ 深さのマージ過程を表す木における部分木
 - $\rightarrow$ オイラーツアー区間

小課題 6

- 各頂点は座標 (x,y) を持っている。
- 回答クエリでは $l_1 \leq x \leq r_1$ かつ $l_2 \leq y \leq r_2$ を満たす点の総和を求めたい。

→長方形領域の総和を求める問題に帰着される

小課題 6

- 回答クエリは最後にまとめてオフライン処理
→ BIT + 平面走査で $O((Q+N)\log(Q+N))$

小課題 7

追加の制約なし

小課題5と小課題6の解法を組み合わせます

AC !

実装は…？

結構
大変です

- 木を2つ構築する
- それぞれオイラーツアー順に並べる
- マージテクをする
- 区間を計算する
- 最後に平面走査
- 加算クエリの実処理も難しい

別解

- 小課題5の解法のボトルネックを思い出す
- 回答クエリで該当する頂点を全部見ていた（最悪 N 回）

→ 回答クエリで現在頂点 A に集まっているビーバーたちは今後同一視できるのでは？

→ set など管理して見る度に erase してまとめてやればいい

実装は…？

40行ぐらい！！

得点分布



100点

55点

42点

38点

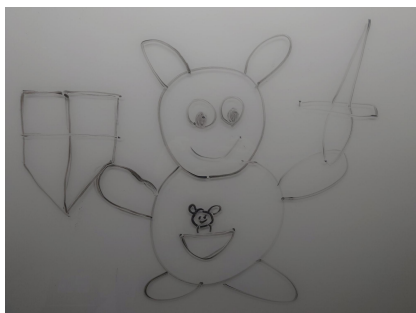
30点

27点

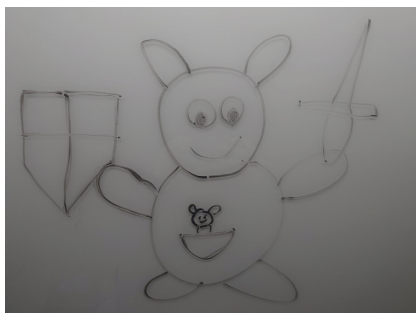
25点

20点

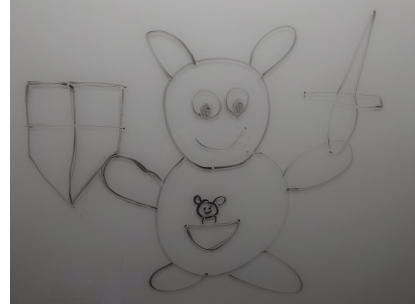
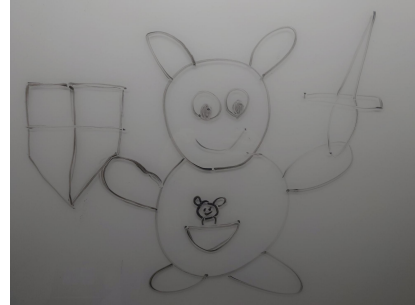
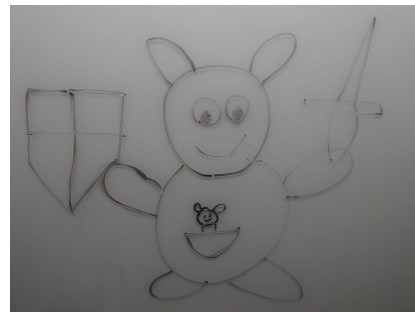
12点



8点



4点



0点