# **Pre-Coding Essentials (Component: crates/vm\_algo/src/tabulation/ranked\_irv.rs, Version/FormulaID: VM-ENGINE v0) — 40/89**

## **1) Goal & Success**

Goal: Deterministically tabulate **IRV** per unit: repeated lowest-elimination, transfers to next **continuing** preference, fixed exhaustion policy, round logs.

Success: Stops when a candidate reaches **majority of continuing ballots** or only one remains; logs eliminations/transfers/exhausted counts; no floats/RNG in tallying.

## **2) Scope**

In scope: Per-unit IRV from ranked ballots (compressed ballot groups), exhaustion policy reduce\_continuing\_denominator, canonical option order, audit log.

Out of scope: Allocation/WTA, gates math, I/O/schema parsing.

## **3) Inputs → Outputs**

Inputs:

ballots: &[(Vec<OptionId>, u64)] (ranking vectors + counts, already validated)

options: &[OptionItem] (to enforce (order\_index, id) order)

params: &Params (reads VM-VAR-001=ranked\_irv, VM-VAR-006)

Outputs:

(UnitScores, IrvLog) where UnitScores.scores holds the **final** round tallies (winner-only or per-option final tallies), plus Turnout carried through.

## **4) Entities/Tables (minimal)**

## **5) Variables (used here)**

## **6) Functions (signatures only)**

rust

CopyEdit

use std::collections::{BTreeMap, BTreeSet};

use vm\_core::{

ids::{UnitId, OptionId},

entities::{Turnout, OptionItem},

variables::Params,

};

pub struct IrvRound { pub eliminated: OptionId, pub transfers: BTreeMap<OptionId, u64>, pub exhausted: u64 }

pub struct IrvLog { pub rounds: Vec<IrvRound>, pub winner: OptionId }

pub fn tabulate\_ranked\_irv(

unit\_id: UnitId,

ballots: &[(Vec<OptionId>, u64)],

options: &[OptionItem],

turnout: Turnout,

params: &Params,

) -> (UnitScores, IrvLog);

## **7) Algorithm Outline (implementation plan)**

**Initialize**

continuing = ordered set of options by (order\_index, id); compute first-preference tallies; continuing\_total = valid\_ballots.

**Majority check**

If some option’s tally > continuing\_total/2, declare winner. “Majority of continuing ballots” per spec.

**Find lowest**

Select lowest tally; break ties **deterministically** by (order\_index, id) (tie policy for IRV eliminations stays deterministic within tabulation; RNG is only for allocation ties per Doc 4B/4C).

**Transfer ballots**

For every group that currently sits on the eliminated option, scan forward to next **continuing** preference; if none, **exhaust** the group. Under policy **reduce\_continuing\_denominator**, subtract exhausted from continuing\_total.

**Log round**

Record IrvRound { eliminated, transfers, exhausted }.

**Repeat** until winner or single continuing option remains.

**Assemble outputs**

UnitScores with final round tallies; IrvLog with rounds and winner.

## **8) State Flow**

Feeds **ALLOCATE** only indirectly for executive-style single-winner contexts (magnitude=1); otherwise, IRV result is the unit winner. Tests demonstrate this workflow.

## **9) Determinism & Numeric Rules**

Stable option order and BTreeMap ensure deterministic iteration.

Integer math only; comparisons follow exact integers; reporting/percent handling is elsewhere; rounding policy pertains to gates/reporting, not IRV tallies.

## **10) Edge Cases & Failure Policy**

**Empty ballots / valid\_ballots=0** ⇒ no majority; winner becomes deterministic smallest (order\_index, id) after eliminations collapse (log zero rounds).

**Ballots with repeats or unknown IDs** ⇒ assume pre-validated by loader; if encountered, skip unknowns within a ballot when seeking next continuing.

**All remaining tied at zero** ⇒ eliminate deterministically until one remains.

**Turnout inconsistency** (e.g., negative counts impossible; exhausted may equal valid\_ballots).

## **11) Test Checklist (must pass)**

**Exhaustion flow** (Annex B Part 3 IRV case) reproduces majority of **continuing** ballots and winner, with correct exhausted count evolution.

Deterministic elimination tie: reorder option IDs of equal tallies → same winner/log due to canonical order.

VM-VAR-006 honored: exhausted ballots reduce the **continuing denominator** exactly as specified.

Gates denominator toggles **do not** affect IRV tallies (only legitimacy checks later).