

# HDCG Study Plan — Full User Stories

One story card per chapter (visual style matches attached examples).

## HDCG-01 — Finite Point Configurations

HDCG-01

### Finite Point Configurations

**Epic / Feature** Combinatorial & Discrete Geometry

**Business Value** Establish theoretical tools and combinatorial principles that underpin later algorithms and applications. (Finite Point Configurations).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** —

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Finite Point Configurations so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Recreate classic extremal/incidence examples; compute bounds on small instances.
- ☐ Generate synthetic point sets (random, grids, clustered) and measure incidence properties.
- ☐ Write a short note contrasting combinatorial vs. metric phenomena observed.

## HDCG-02 — Packing and Covering

HDCG-02

Packing and Covering

**Epic / Feature** Combinatorial & Discrete Geometry

**Business Value** Establish theoretical tools and combinatorial principles that underpin later algorithms and applications. (Packing and Covering).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Packing and Covering so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Derive density bounds for circle/sphere packing in small domains; validate by experiment.
- ☐ Implement a greedy packing heuristic; compare to known optimal layouts for toy sizes.
- ☐ Compute simple covering numbers for intervals/disks; visualize uncovered mass.

## HDCG-03 — Tilings

HDCG-03

Tilings

**Epic / Feature** Combinatorial & Discrete Geometry

**Business Value** Establish theoretical tools and combinatorial principles that underpin later algorithms and applications. (Tilings).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Tilings so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Classify tilings (periodic vs. aperiodic) for given prototiles; prove a simple property.
- ☐ Implement a substitution tiling generator; render several levels of refinement.
- ☐ Measure tile frequency and boundary growth; summarize findings.

## HDCG-04 — Helly-type Theorems & Transversals

HDCG-04

### Helly-type Theorems & Transversals

**Epic / Feature** Combinatorial & Discrete Geometry

**Business Value** Establish theoretical tools and combinatorial principles that underpin later algorithms and applications. (Helly-type Theorems & Transversals).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Helly-type Theorems & Transversals so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ State Helly/Carathéodory/Tverberg precisely; prove a concrete low-dimensional instance.
- ☐ Model a convex-feasibility LP; empirically test Helly-style certificates of feasibility.
- ☐ Create counterexamples to naive generalizations; document assumptions.

## HDCG-05 — Pseudoline Arrangements

HDCG-05

### Pseudoline Arrangements

**Epic / Feature** Combinatorial & Discrete Geometry

**Business Value** Establish theoretical tools and combinatorial principles that underpin later algorithms and applications. (Pseudoline Arrangements).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Pseudoline Arrangements so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/allly checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Draw small arrangements; compute cells/levels/zones and verify counts.
- ☐ Implement arrangement construction for segments/lines and enumerate faces.
- ☐ Explore zone theorem numerically by measuring average zone complexity.

## HDCG-06 — Oriented Matroids

HDCG-06

Oriented Matroids

**Epic / Feature** Combinatorial & Discrete Geometry

**Business Value** Establish theoretical tools and combinatorial principles that underpin later algorithms and applications. (Oriented Matroids).

**Priority / Estimate** **Priority:** Must

**SP: 3**

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Oriented Matroids so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Work with sign vectors and chirotopes on small point sets; verify axioms.
- ☐ Relate realizable vs. non-realizable examples; find and reproduce a literature example.
- ☐ Map an arrangement to an oriented matroid; note dualities.

## HDCG-07 — Lattice Points & Lattice Polytopes

HDCG-07

### Lattice Points & Lattice Polytopes

**Epic / Feature** Combinatorial & Discrete Geometry

**Business Value** Establish theoretical tools and combinatorial principles that underpin later algorithms and applications. (Lattice Points & Lattice Polytopes).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Lattice Points & Lattice Polytopes so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Compute Ehrhart polynomials for small lattice polytopes; verify reciprocity numerically.
- ☐ Implement lattice-point counting for boxes/simplices; validate against closed forms.
- ☐ Investigate how dilation changes counts and coefficients.

## HDCG-08 — Low-Distortion Embeddings

HDCG-08

Low-Distortion Embeddings

**Epic / Feature** Combinatorial & Discrete Geometry

**Business Value** Establish theoretical tools and combinatorial principles that underpin later algorithms and applications. (Low-Distortion Embeddings).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Low-Distortion Embeddings so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement a Johnson–Lindenstrauss (JL) embedding; measure distortion for varying  $k$ .
- ☐ Compare PCA vs. random projections on a real dataset; report reconstruction error.
- ☐ Document trade-offs (run time, memory, accuracy) across embedding choices.

## HDCG-09 — Polygonal Linkages

HDCG-09

Polygonal Linkages

**Epic / Feature** Combinatorial & Discrete Geometry

**Business Value** Establish theoretical tools and combinatorial principles that underpin later algorithms and applications. (Polygonal Linkages).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Polygonal Linkages so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Simulate a simple linkage; visualize configuration space qualitatively.
- ☐ Test feasibility (realisability) for a small linkage with constraints.
- ☐ Identify singular configurations and discuss rigidity vs. flexibility.

## HDCG-10 — Geometric Graph Theory

HDCG-10

### Geometric Graph Theory

**Epic / Feature** Combinatorial & Discrete Geometry

**Business Value** Establish theoretical tools and combinatorial principles that underpin later algorithms and applications. (Geometric Graph Theory).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Geometric Graph Theory so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Compute visibility graphs and planarity tests on random point sets.
- ☐ Measure crossing numbers experimentally for small  $n$ ; compare to bounds.
- ☐ Explore thickness and minors on selected graphs.

## HDCG-11 — Euclidean Ramsey Theory

HDCG-11

### Euclidean Ramsey Theory

**Epic / Feature** Combinatorial & Discrete Geometry

**Business Value** Establish theoretical tools and combinatorial principles that underpin later algorithms and applications. (Euclidean Ramsey Theory).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Euclidean Ramsey Theory so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement colorings for points/edges; search for monochromatic structures.
- ☐ Reproduce a small Euclidean Ramsey statement; provide a constructive or probabilistic proof sketch.
- ☐ Summarize growth rates and open directions.

## HDCG-12 — Discrete Aspects of Stochastic Geometry

HDCG-12

### Discrete Aspects of Stochastic Geometry

**Epic / Feature** Combinatorial & Discrete Geometry

**Business Value** Establish theoretical tools and combinatorial principles that underpin later algorithms and applications. (Discrete Aspects of Stochastic Geometry).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth; variance in experiments; need for many trials

*As a research student of discrete geometry, I want to master Discrete Aspects of Stochastic Geometry so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Simulate Poisson point processes; estimate mean area/length of derived structures.
- ☐ Compute mean widths/coverage via Monte Carlo; include confidence intervals.
- ☐ Compare empirical findings to theoretical expectations where available.

## HDCG-13 — Geometric Discrepancy & Uniform Distribution

HDCG-13

### Geometric Discrepancy & Uniform Distribution

**Epic / Feature** Combinatorial & Discrete Geometry

**Business Value** Establish theoretical tools and combinatorial principles that underpin later algorithms and applications. (Geometric Discrepancy & Uniform Distribution).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Geometric Discrepancy & Uniform Distribution so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/allly checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Compute discrepancy of random vs. low-discrepancy sequences (Halton/Sobol) on test sets.
- ☐ Integrate a smooth test function using QMC vs. MC; compare convergence rates.
- ☐ Relate discrepancy to VC-dimension or range spaces from the chapter.

## HDCG-14 — Polyominoes

HDCG-14

Polyominoes

**Epic / Feature** Combinatorial & Discrete Geometry

**Business Value** Establish theoretical tools and combinatorial principles that underpin later algorithms and applications. (Polyominoes).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Polyominoes so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Use exact cover (DLX) to tile small regions; verify parity and area constraints.
- ☐ Classify tilings for a given board; produce enumerations for small sizes.
- ☐ Document invariants that quickly rule out impossible tilings.

## HDCG-15 — Convex Polytopes: Basics

HDCG-15

Convex Polytopes: Basics

**Epic / Feature** Polytopes & Polyhedra

**Business Value** Build fluency with polytopal structures used across optimization, geometry processing, and combinatorics. (Convex Polytopes: Basics).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Convex Polytopes: Basics so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/allly checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Convert small polytopes between H- and V-representations; check duals.
- ☐ Illustrate faces/facets/normal fans with a plotting tool; verify Euler.
- ☐ Summarize duality relationships with examples (simplex, cube, cross-polytope).

## HDCG-16 — Subdivisions & Triangulations of Polytopes

HDCG-16

### Subdivisions & Triangulations of Polytopes

**Epic / Feature** Polytopes & Polyhedra

**Business Value** Build fluency with polytopal structures used across optimization, geometry processing, and combinatorics. (Subdivisions & Triangulations of Polytopes).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced, convex hulls, orientation/circumcircle predicates

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Subdivisions & Triangulations of Polytopes so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility Performance Scalability

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Construct regular triangulations; implement flips and track secondary polytopes qualitatively.
- ☐ Evaluate simplex quality metrics (aspect ratio, minimum angle).
- ☐ Show effect of lifting on triangulation regularity.

## HDCG-17 — Face Numbers: $f$ -, $h$ -, and $g$ -vectors

HDCG-17

Face Numbers:  $f$ -,  $h$ -, and  $g$ -vectors

**Epic / Feature** Polytopes & Polyhedra

**Business Value** Build fluency with polytopal structures used across optimization, geometry processing, and combinatorics. (Face Numbers:  $f$ -,  $h$ -, and  $g$ -vectors).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Face Numbers:  $f$ -,  $h$ -, and  $g$ -vectors so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Compute  $f$ - and  $h$ -vectors for classic polytopes; check Dehn–Sommerville relations.
- ☐ Explore  $g$ -theorem examples; illustrate inequalities with plots.
- ☐ Prepare a cheat sheet summarizing identities and constraints.

## HDCG-18 — Symmetry of Polytopes

HDCG-18

Symmetry of Polytopes

**Epic / Feature** Polytopes & Polyhedra

**Business Value** Build fluency with polytopal structures used across optimization, geometry processing, and combinatorics. (Symmetry of Polytopes).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Symmetry of Polytopes so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/allly checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Identify automorphism groups for basic polytopes; compute group orders.
- ☐ Relate symmetry to orbits of faces; visualize symmetric embeddings.
- ☐ Discuss symmetry exploitation in algorithms (state-space reduction).

## HDCG-19 — Polytope Skeletons & Paths

HDCG-19

### Polytope Skeletons & Paths

**Epic / Feature** Polytopes & Polyhedra

**Business Value** Build fluency with polytopal structures used across optimization, geometry processing, and combinatorics. (Polytope Skeletons & Paths).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Polytope Skeletons & Paths so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all1y checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Experiment with graph diameters; simulate pivot paths on polytope graphs.
- ☐ Implement simple pivot rules; measure paths vs. diameter bounds.
- ☐ Relate results to Hirsch-type questions qualitatively.

## HDCG-20 — Polyhedral Maps

HDCG-20

Polyhedral Maps

**Epic / Feature** Polytopes & Polyhedra

**Business Value** Build fluency with polytopal structures used across optimization, geometry processing, and combinatorics. (Polyhedral Maps).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Polyhedral Maps so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability    Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/allly checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Embed simple polyhedral maps on surfaces; verify Euler characteristic.
- ☐ Count faces/edges/vertices; confirm orientability effects.
- ☐ Construct a small map with specified degree sequence.

## HDCG-21 — Topological Methods in Discrete Geometry

HDCG-21

### Topological Methods in Discrete Geometry

**Epic / Feature** Combinatorial & Computational Topology

**Business Value** Leverage topological perspectives to reason about existence, structure, and invariants in geometric problems. (Topological Methods in Discrete Geometry).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Topological Methods in Discrete Geometry so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Use ham-sandwich or Borsuk–Ulam style arguments to prove a discrete claim.
- ☐ Illustrate topological proof vs. combinatorial alternative on the same problem.
- ☐ Document assumptions and generalization limits.

## HDCG-22 — Random Simplicial Complexes

HDCG-22

### Random Simplicial Complexes

**Epic / Feature** Combinatorial & Computational Topology

**Business Value** Leverage topological perspectives to reason about existence, structure, and invariants in geometric problems. (Random Simplicial Complexes).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth; variance in experiments; need for many trials

*As a research student of discrete geometry, I want to master Random Simplicial Complexes so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Generate Linial–Meshulam or Vietoris–Rips models; sweep probability parameters.
- ☐ Measure thresholds for connectedness or vanishing homology.
- ☐ Plot Betti numbers across regimes; discuss finite-size effects.

## HDCG-23 — Graphs on Surfaces (Embeddings & Genus)

HDCG-23

### Graphs on Surfaces (Embeddings & Genus)

**Epic / Feature** Combinatorial & Computational Topology

**Business Value** Leverage topological perspectives to reason about existence, structure, and invariants in geometric problems. (Graphs on Surfaces (Embeddings & Genus)).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth; numerical robustness and degeneracies

*As a research student of discrete geometry, I want to master Graphs on Surfaces (Embeddings & Genus) so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility Robustness Accuracy

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement planarity testing; compute genus for small graphs.
- ☐ Route edges on a surface embedding; visualize crossings by handles.
- ☐ Compare embeddings with/without constraints.

## HDCG-24 — Persistent Homology (Barcodes)

HDCG-24

### Persistent Homology (Barcodes)

**Epic / Feature** Combinatorial & Computational Topology

**Business Value** Leverage topological perspectives to reason about existence, structure, and invariants in geometric problems. (Persistent Homology (Barcodes)).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** research student of discrete geometry

**Dependencies** prior chapters as referenced, simplicial complexes, metrics, stability notion

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a research student of discrete geometry, I want to master Persistent Homology (Barcodes) so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/allly checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Build filtrations (Rips/Čech); compute 0–2D barcodes on a toy dataset.
- ☐ Study stability by adding noise and comparing bottleneck distance.
- ☐ Interpret features (lifetimes) with domain context.

## HDCG-25 — High-dimensional Topological Data Analysis

HDCG-25

### High-dimensional Topological Data Analysis

**Epic / Feature** Combinatorial & Computational Topology

**Business Value** Leverage topological perspectives to reason about existence, structure, and invariants in geometric problems. (High-dimensional Topological Data Analysis).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** data scientist working with geometric methods

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth; curse of dimensionality; memory pressure

*As a data scientist working with geometric methods, I want to master High-dimensional Topological Data Analysis so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Construct sparse filtrations; apply landmarks/witness complexes.
- ☐ Use dimensionality reduction to visualize summaries; validate with metrics.
- ☐ Evaluate scalability (time/memory) vs. sample size and dimension.

## HDCG-26 — Convex Hulls

HDCG-26

Convex Hulls

**Epic / Feature** Algorithms of Fundamental Geometric Objects

**Business Value** Master core geometric algorithms that power search, reconstruction, meshing, and planning pipelines. (Convex Hulls).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master Convex Hulls so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility Performance Scalability

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement 2D (Andrew) and 3D (incremental) hulls; include degeneracy handling.
- ☐ Add exact/filtered predicates; benchmark accuracy and speed.
- ☐ Validate by area/volume/facet orientation; export meshes.

## HDCG-27 — Voronoi Diagrams & Delaunay Triangulations

HDCG-27

### Voronoi Diagrams & Delaunay Triangulations

**Epic / Feature** Algorithms of Fundamental Geometric Objects

**Business Value** Master core geometric algorithms that power search, reconstruction, meshing, and planning pipelines. (Voronoi Diagrams & Delaunay Triangulations).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced, convex hulls, orientation/circumcircle predicates

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master Voronoi Diagrams & Delaunay Triangulations so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility Performance Scalability

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Build Delaunay via edge flips/incremental insertion; output Voronoi by duality.
- ☐ Implement point location; compare query latency to kd-tree baseline.
- ☐ Stress-test co-circular/duplicate inputs; enable exact predicates.

## HDCG-28 — Arrangements of Curves and Surfaces

HDCG-28

### Arrangements of Curves and Surfaces

**Epic / Feature** Algorithms of Fundamental Geometric Objects

**Business Value** Master core geometric algorithms that power search, reconstruction, meshing, and planning pipelines. (Arrangements of Curves and Surfaces).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth; numerical robustness and degeneracies

*As a applied geometry engineer, I want to master Arrangements of Curves and Surfaces so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility Robustness Accuracy

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Construct arrangements of segments/lines; enumerate cells and graph structure.
- ☐ Measure zone/level complexity empirically; compare to theory.
- ☐ Demonstrate applications (motion planning cells / point location).

## HDCG-29 — Triangulations & Mesh Generation

HDCG-29

### Triangulations & Mesh Generation

**Epic / Feature** Algorithms of Fundamental Geometric Objects

**Business Value** Master core geometric algorithms that power search, reconstruction, meshing, and planning pipelines. (Triangulations & Mesh Generation).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced, convex hulls, orientation/circumcircle predicates

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master Triangulations & Mesh Generation so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability   Reproducibility   Performance   Scalability

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement Delaunay refinement; track minimum angle and element size.
- ☐ Perform boundary recovery; assess quality metrics before/after refinement.
- ☐ Export mesh and run a simple PDE/graphics demo.

## HDCG-30 — Polygons (Geometry & Algorithms)

HDCG-30

### Polygons (Geometry & Algorithms)

**Epic / Feature** Algorithms of Fundamental Geometric Objects

**Business Value** Master core geometric algorithms that power search, reconstruction, meshing, and planning pipelines. (Polygons (Geometry & Algorithms)).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master Polygons (Geometry & Algorithms) so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability   Reproducibility   Performance   Scalability

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement point-in-polygon (ray casting / winding number); triangulate simple polygons.
- ☐ Handle degeneracies (collinearity, repeated vertices); compute area/centroid.
- ☐ Demonstrate art-gallery style visibility on floorplan polygons.

## HDCG-31 — Shortest Paths & Networks

HDCG-31

### Shortest Paths & Networks

**Epic / Feature** Algorithms of Fundamental Geometric Objects

**Business Value** Master core geometric algorithms that power search, reconstruction, meshing, and planning pipelines. (Shortest Paths & Networks).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master Shortest Paths & Networks so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/allly checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Build a visibility graph and run Dijkstra for polygonal domains.
- ☐ Implement continuous Dijkstra or funnel algorithm on triangulations.
- ☐ Compare path lengths and runtimes across methods.

## HDCG-32 — Proximity Algorithms

HDCG-32

### Proximity Algorithms

**Epic / Feature** Algorithms of Fundamental Geometric Objects

**Business Value** Master core geometric algorithms that power search, reconstruction, meshing, and planning pipelines. (Proximity Algorithms).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master Proximity Algorithms so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility Performance Scalability

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/allly checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Construct MST/ $\beta$ -skeletons and k-NN graphs; profile runtimes.
- ☐ Analyse stability under noise; compare exact vs. approximate structures.
- ☐ Summarize use-cases (clustering, skeletonization).

## HDCG-33 — Visibility & Art-Gallery Problems

HDCG-33

### Visibility & Art-Gallery Problems

**Epic / Feature** Algorithms of Fundamental Geometric Objects

**Business Value** Master core geometric algorithms that power search, reconstruction, meshing, and planning pipelines. (Visibility & Art-Gallery Problems).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master Visibility & Art-Gallery Problems so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Compute visibility polygons; handle holes and reflex vertices.
- ☐ Formulate simple art-gallery coverage and test heuristics.
- ☐ Visualize guard placements and uncovered regions.

## HDCG-34 — Geometric Reconstruction Problems

HDCG-34

### Geometric Reconstruction Problems

**Epic / Feature** Algorithms of Fundamental Geometric Objects

**Business Value** Master core geometric algorithms that power search, reconstruction, meshing, and planning pipelines. (Geometric Reconstruction Problems).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master Geometric Reconstruction Problems so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/allly checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Recover shapes from projections or shadows; study minimal measurement sets.
- ☐ Quantify reconstruction error (Hausdorff/symmetric difference) on synthetic data.
- ☐ Discuss identifiability conditions and failure modes.

## HDCG-35 — Curve & Surface Reconstruction

HDCG-35

### Curve & Surface Reconstruction

**Epic / Feature** Algorithms of Fundamental Geometric Objects

**Business Value** Master core geometric algorithms that power search, reconstruction, meshing, and planning pipelines. (Curve & Surface Reconstruction).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth; numerical robustness and degeneracies

*As a applied geometry engineer, I want to master Curve & Surface Reconstruction so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability   Reproducibility   Robustness   Accuracy

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement a simple crust/ball-pivoting/Poisson pipeline on noisy samples.
- ☐ Tune parameters and measure Hausdorff distance to ground truth.
- ☐ Report topology errors and smoothing trade-offs.

## HDCG-36 — Computational Convexity

HDCG-36

### Computational Convexity

**Epic / Feature** Algorithms of Fundamental Geometric Objects

**Business Value** Master core geometric algorithms that power search, reconstruction, meshing, and planning pipelines. (Computational Convexity).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master Computational Convexity so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Solve membership/separation via cutting-plane or ellipsoid on toy instances.
- ☐ Compare oracle-based methods vs. explicit H/V representations.
- ☐ Document complexity and numerical behavior.

## HDCG-37 — Algorithmic Real Algebraic Geometry

HDCG-37

### Algorithmic Real Algebraic Geometry

**Epic / Feature** Algorithms of Fundamental Geometric Objects

**Business Value** Master core geometric algorithms that power search, reconstruction, meshing, and planning pipelines. (Algorithmic Real Algebraic Geometry).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth; numerical robustness and degeneracies

*As a applied geometry engineer, I want to master Algorithmic Real Algebraic Geometry so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility Robustness Accuracy

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Experiment with solving small semi-algebraic systems; visualize solution sets.
- ☐ Use cylindrical algebraic decomposition (CAD) conceptually or via a CAS for toy inputs.
- ☐ Discuss complexity blowups and practical workarounds.

## HDCG-38 — Point Location

HDCG-38

Point Location

**Epic / Feature** Geometric Data Structures & Searching

**Business Value** Develop data structures for fast queries, intersections, and proximity at scale. (Point Location).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master Point Location so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility Performance Scalability

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Build a randomized trapezoidal map; benchmark query vs. build time.
- ☐ Test degeneracies and dynamic updates (insertions).
- ☐ Compare to persistent search structures where applicable.

## HDCG-39 — Collision Detection & Proximity Queries

HDCG-39

### Collision Detection & Proximity Queries

**Epic / Feature** Geometric Data Structures & Searching

**Business Value** Develop data structures for fast queries, intersections, and proximity at scale. (Collision Detection & Proximity Queries).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth; numerical robustness and degeneracies

*As a applied geometry engineer, I want to master Collision Detection & Proximity Queries so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility Robustness Accuracy

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement BVH (AABB/OBB) construction and traversal; add narrow-phase tests.
- ☐ Evaluate continuous collision detection for moving segments/triangles.
- ☐ Profiling: queries/sec vs. object count; document worst-case scenes.

## HDCG-40 — Range Searching

HDCG-40

Range Searching

**Epic / Feature** Geometric Data Structures & Searching

**Business Value** Develop data structures for fast queries, intersections, and proximity at scale. (Range Searching).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master Range Searching so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability   Reproducibility   Performance   Scalability

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/allly checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement kd/interval/segment trees; test orthogonal range counting/reporting.
- ☐ Measure query/update trade-offs; visualize pruning behavior.
- ☐ Scale experiments to large  $n$  and report memory footprints.

## HDCG-41 — Ray Shooting & Lines in Space

HDCG-41

### Ray Shooting & Lines in Space

**Epic / Feature** Geometric Data Structures & Searching

**Business Value** Develop data structures for fast queries, intersections, and proximity at scale. (Ray Shooting & Lines in Space).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master Ray Shooting & Lines in Space so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability   Reproducibility   Performance   Scalability

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement ray-scene intersection with uniform grid and BVH; compare.
- ☐ Validate against analytic scenes; collect miss/hit statistics.
- ☐ Profile coherent vs. incoherent rays.

## HDCG-42 — Geometric Intersection

HDCG-42

Geometric Intersection

**Epic / Feature** Geometric Data Structures & Searching

**Business Value** Develop data structures for fast queries, intersections, and proximity at scale. (Geometric Intersection).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth; numerical robustness and degeneracies

*As a applied geometry engineer, I want to master Geometric Intersection so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility Performance Scalability

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/allly checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Write exact segment/triangle intersection predicates; fuzz edge cases.
- ☐ Build sweep-line for segment intersection; report complexity and statistics.
- ☐ Summarize robustness fixes (epsilon vs. exact arithmetic).

## HDCG-43 — Nearest Neighbors in High Dimension

HDCG-43

### Nearest Neighbors in High Dimension

**Epic / Feature** Geometric Data Structures & Searching

**Business Value** Develop data structures for fast queries, intersections, and proximity at scale. (Nearest Neighbors in High Dimension).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth; curse of dimensionality; memory pressure

*As a applied geometry engineer, I want to master Nearest Neighbors in High Dimension so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility Performance Scalability

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Compare kd-tree, LSH, and HNSW on real vector data; tune parameters.
- ☐ Measure recall/latency trade-offs; draw accuracy-speed curves.
- ☐ Discuss curse-of-dimensionality and mitigation strategies.

## HDCG-44 — Randomization & Derandomization in Geometry

HDCG-44

### Randomization & Derandomization in Geometry

**Epic / Feature** Computational Techniques

**Business Value** Adopt practical computation techniques for speed, robustness, generalization, and summaries. (Randomization & Derandomization in Geometry).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth; variance in experiments; need for many trials

*As a applied geometry engineer, I want to master Randomization & Derandomization in Geometry so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement random sampling/ $\epsilon$ -nets for a range space; validate bounds empirically.
- ☐ Replace with a deterministic construction (derandomization) on small cases.
- ☐ Compare quality and runtime for both approaches.

## HDCG-45 — Robust Geometric Computation

HDCG-45

### Robust Geometric Computation

**Epic / Feature** Computational Techniques

**Business Value** Adopt practical computation techniques for speed, robustness, generalization, and summaries. (Robust Geometric Computation).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth; numerical robustness and degeneracies

*As a applied geometry engineer, I want to master Robust Geometric Computation so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability   Reproducibility   Robustness   Accuracy

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Integrate exact/filtered predicates into one pipeline; catalog failures with floats.
- ☐ Re-run earlier algorithms (hull, Delaunay, intersections) under robustness modes.
- ☐ Summarize cost of robustness vs. correctness benefits.

## HDCG-46 — Parallel Algorithms in Geometry

HDCG-46

### Parallel Algorithms in Geometry

**Epic / Feature** Computational Techniques

**Business Value** Adopt practical computation techniques for speed, robustness, generalization, and summaries. (Parallel Algorithms in Geometry).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced, task parallelism basics, threading model

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master Parallel Algorithms in Geometry so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility Performance Scalability

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Parallelize a hull or Delaunay implementation; measure speedup and scalability.
- ☐ Identify contention hotspots; propose work partitioning.
- ☐ Test on varying cores; produce a scalability plot.

## HDCG-47 — $\varepsilon$ -nets & $\varepsilon$ -approximations

HDCG-47

$\varepsilon$ -nets &  $\varepsilon$ -approximations

**Epic / Feature** Computational Techniques

**Business Value** Adopt practical computation techniques for speed, robustness, generalization, and summaries. ( $\varepsilon$ -nets &  $\varepsilon$ -approximations).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master  $\varepsilon$ -nets &  $\varepsilon$ -approximations so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/allly checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Construct  $\varepsilon$ -nets/approximations for basic range spaces; verify hitting/approximation properties.
- ☐ Relate to VC-dimension; compute sample sizes for target  $\varepsilon$ ,  $\delta$ .
- ☐ Apply to a small learning-like problem (set cover/active sampling).

## HDCG-48 — Coresets & Sketches

HDCG-48

Coresets & Sketches

**Epic / Feature** Computational Techniques

**Business Value** Adopt practical computation techniques for speed, robustness, generalization, and summaries. (Coresets & Sketches).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master Coresets & Sketches so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility Performance Scalability

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Build k-means/median coresets; evaluate clustering error vs. exact.
- ☐ Profile construction time vs. coreset size; plot trade-offs.
- ☐ Demonstrate downstream speedups with negligible loss.

## HDCG-49 — Linear Programming (Low-dimensional & Randomized)

HDCG-49		Linear Programming (Low-dimensional & Randomized)			
Epic / Feature	Applications of Discrete & Computational Geometry				
Business Value	Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Linear Programming (Low-dimensional & Randomized)).				
Priority / Estimate	Priority: Must			SP: 3	
Persona	practitioner building geometry-driven applications				
Dependencies	prior chapters as referenced				
Assumptions / Risks	time to internalize proofs vs. breadth; variance in experiments; need for many trials				
As a practitioner building geometry-driven applications, I want to master Linear Programming (Low-dimensional & Randomized) so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.					
Non-Functional	Reliability	Reproducibility	Performance	Scalability	
Acceptance Criteria (BDD)					
Scenario	Happy path				
Given	the chapter, examples, and any tooling are available				
When	I complete the <i>Hands-on Objective</i> and validations for this chapter				
Then	the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks				
Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • Definition of Done: All ACs pass; Tests green; Security/allly checks; Docs updated; Deployed/flagged.					

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement randomized incremental LP in low dimension; visualize feasible region.
- ☐ Benchmark vs. simplex on small test sets; record degeneracy behavior.
- ☐ Apply to a geometric optimization mini-problem (smallest enclosing ball).

## HDCG-50 — Algorithmic Motion Planning

HDCG-50

### Algorithmic Motion Planning

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Algorithmic Motion Planning).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** practitioner building geometry-driven applications

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a practitioner building geometry-driven applications, I want to master Algorithmic Motion Planning so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement PRM/RRT on a 2D environment; measure coverage and path quality.
- ☐ Add collision checks via BVH; compare planners across seeds.
- ☐ Export paths and visualize milestones/edges.

## HDCG-51 — Robotics: Configuration Spaces

HDCG-51

Robotics: Configuration Spaces

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Robotics: Configuration Spaces).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** practitioner building geometry-driven applications

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a practitioner building geometry-driven applications, I want to master Robotics: Configuration Spaces so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability    Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Model a planar arm's configuration space with obstacles; compute free space components.
- ☐ Plan a collision-free path; validate in a simple simulator.
- ☐ Discuss DOF scaling and sampling strategies.

## HDCG-52 — Computer Graphics: Geometric Pipelines

HDCG-52

### Computer Graphics: Geometric Pipelines

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Computer Graphics: Geometric Pipelines).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** practitioner building geometry-driven applications

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a practitioner building geometry-driven applications, I want to master Computer Graphics: Geometric Pipelines so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability   Reproducibility   Usability

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement geometric clipping and rasterization of a simple scene.
- ☐ Run mesh simplification; report quality vs. decimation.
- ☐ Profile the pipeline stages you implemented.

## HDCG-53 — Modeling Motion (Rigid & Affine)

HDCG-53

Modeling Motion (Rigid & Affine)

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Modeling Motion (Rigid & Affine)).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** practitioner building geometry-driven applications

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a practitioner building geometry-driven applications, I want to master Modeling Motion (Rigid & Affine) so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability    Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/allly checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Derive rigid/affine transforms; implement screw interpolation demo.
- ☐ Track errors under concatenation; verify invariants.
- ☐ Compare different interpolation schemes for stability.

## HDCG-54 — Pattern Recognition (Geometric View)

HDCG-54

### Pattern Recognition (Geometric View)

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Pattern Recognition (Geometric View)).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** practitioner building geometry-driven applications

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a practitioner building geometry-driven applications, I want to master Pattern Recognition (Geometric View) so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability    Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/allly checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement geometric classifiers (nearest-center/Voronoi) on a toy dataset.
- ☐ Compare to an SVM baseline; report decision boundary shapes.
- ☐ Analyze robustness to outliers using geometric medians.

## HDCG-55 — Graph Drawing

HDCG-55

Graph Drawing

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Graph Drawing).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** practitioner building geometry-driven applications

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a practitioner building geometry-driven applications, I want to master Graph Drawing so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability   Reproducibility   Usability

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement force-directed layout with planarity constraints where possible.
- ☐ Test layered/Sugiyama layout; manage crossings.
- ☐ Quantify edge length variance and crossing counts.

## HDCG-56 — Splines & Geometric Modeling

HDCG-56

### Splines & Geometric Modeling

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Splines & Geometric Modeling).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** practitioner building geometry-driven applications

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a practitioner building geometry-driven applications, I want to master Splines & Geometric Modeling so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement B-spline/NURBS evaluation; verify C1/C2 continuity on examples.
- ☐ Fit curves to sample points; measure error sensitivity.
- ☐ Render and annotate control polygon effects.

## HDCG-57 — Solid Modeling (B-Rep & CSG)

HDCG-57

### Solid Modeling (B-Rep & CSG)

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Solid Modeling (B-Rep & CSG)).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** practitioner building geometry-driven applications

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth; numerical robustness and degeneracies

*As a practitioner building geometry-driven applications, I want to master Solid Modeling (B-Rep & CSG) so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility Robustness Accuracy

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Convert between CSG and B-rep for simple solids; implement Boolean ops.
- ☐ Detect and repair non-manifold issues; validate watertightness.
- ☐ Export to a CAD-friendly format and roundtrip.

## HDCG-58 — Robust Statistics: Data Depth & Medians

HDCG-58

### Robust Statistics: Data Depth & Medians

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Robust Statistics: Data Depth & Medians).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** data scientist working with geometric methods

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth; numerical robustness and degeneracies

*As a data scientist working with geometric methods, I want to master Robust Statistics: Data Depth & Medians so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability   Reproducibility   Robustness   Accuracy

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Compute Tukey depth and halfspace medians on 2D data; visualize contours.
- ☐ Compare robust vs. least-squares fits under outliers.
- ☐ Report breakdown points and runtime.

## HDCG-59 — Geographic Information Systems (GIS)

HDCG-59

### Geographic Information Systems (GIS)

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Geographic Information Systems (GIS)).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** practitioner building geometry-driven applications

**Dependencies** prior chapters as referenced, map projections, spherical geometry basics

**Assumptions / Risks** time to internalize proofs vs. breadth; numerical robustness and degeneracies

*As a practitioner building geometry-driven applications, I want to master Geographic Information Systems (GIS) so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility Robustness Accuracy

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Implement point-in-polygon on geodesic coordinates; handle holes and winding.
- ☐ Project between WGS84 and a planar CRS; quantify distortion.
- ☐ Run spatial joins at scale; profile performance.

## HDCG-60 — Grassmann–Cayley Algebra

HDCG-60

### Grassmann–Cayley Algebra

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Grassmann–Cayley Algebra).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** practitioner building geometry-driven applications

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a practitioner building geometry-driven applications, I want to master Grassmann–Cayley Algebra so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Represent joins/meets using Grassmann–Cayley algebra; solve a projective incidence problem.
- ☐ Work a small numerical example to verify identities.
- ☐ Document where this algebra simplifies proofs.

## HDCG-61 — Rigidity & Scene Analysis

HDCG-61

### Rigidity & Scene Analysis

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Rigidity & Scene Analysis).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** practitioner building geometry-driven applications

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a practitioner building geometry-driven applications, I want to master Rigidity & Scene Analysis so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Test bar-and-joint rigidity via rank conditions; visualize stresses.
- ☐ Generate minimally rigid (Laman) graphs; perturb to test stability.
- ☐ Relate rigidity to structure-from-motion intuition.

## HDCG-62 — Rigidity of Symmetric Frameworks

HDCG-62

### Rigidity of Symmetric Frameworks

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Rigidity of Symmetric Frameworks).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** practitioner building geometry-driven applications

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a practitioner building geometry-driven applications, I want to master Rigidity of Symmetric Frameworks so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Use group actions to adapt rigidity counts; build symmetric examples.
- ☐ Check how symmetry changes generic rigidity; demonstrate a case study.
- ☐ Summarize computational implications.

## HDCG-63 — Global Rigidity

HDCG-63

Global Rigidity

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Global Rigidity).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** practitioner building geometry-driven applications

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a practitioner building geometry-driven applications, I want to master Global Rigidity so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Differentiate local vs. global rigidity; check generic conditions on small graphs.
- ☐ Search for counterexamples; visualize multiple embeddings.
- ☐ Note algorithmic challenges and complexity hints.

## HDCG-64 — Crystals: Periodic & Aperiodic Structures

HDCG-64

### Crystals: Periodic & Aperiodic Structures

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Crystals: Periodic & Aperiodic Structures).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** practitioner building geometry-driven applications

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a practitioner building geometry-driven applications, I want to master Crystals: Periodic & Aperiodic Structures so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability    Reproducibility

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Generate periodic nets; compute basic invariants and visualize unit cells.
- ☐ Contrast with an aperiodic example; discuss diffraction-like signatures.
- ☐ Explore stability under perturbations.

## HDCG-65 — Structural Molecular Biology (Distance Geometry)

HDCG-65	Structural Molecular Biology (Distance Geometry)			
Epic / Feature	Applications of Discrete & Computational Geometry			
Business Value	Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Structural Molecular Biology (Distance Geometry)).			
Priority / Estimate	Priority: Must	SP: 3		
Persona	practitioner building geometry-driven applications			
Dependencies	prior chapters as referenced			
Assumptions / Risks	time to internalize proofs vs. breadth; numerical robustness and degeneracies			
As a practitioner building geometry-driven applications, I want to master Structural Molecular Biology (Distance Geometry) so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.				
Non-Functional	Reliability	Reproducibility	Robustness	Accuracy
Acceptance Criteria (BDD)				
Scenario	Happy path			
Given	the chapter, examples, and any tooling are available			
When	I complete the <i>Hands-on Objective</i> and validations for this chapter			
Then	the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks			
Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • Definition of Done: All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.				

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Solve a small distance-geometry problem for a peptide backbone; reconstruct coordinates.
- ☐ Quantify reconstruction error under noise and missing distances.
- ☐ Discuss ambiguities and constraints (chirality, bond lengths).

## HDCG-66 — Geometry & Topology of Genomic Data

HDCG-66

### Geometry & Topology of Genomic Data

**Epic / Feature** Applications of Discrete & Computational Geometry

**Business Value** Apply geometric concepts to real domains to deliver measurable outcomes and demos. (Geometry & Topology of Genomic Data).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** data scientist working with geometric methods

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a data scientist working with geometric methods, I want to master Geometry & Topology of Genomic Data so that I can apply it to real problems and communicate theoretical insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

#### Acceptance Criteria (BDD)

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

#### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Embed genomic relationships using geometric or topological summaries.
- ☐ Compare metrics/embeddings (edit distance, Hamming, phylogenetic).
- ☐ Interpret structures in terms of recombination/phylogeny signals.

## HDCG-67 — Geometric Software (Survey)

HDCG-67

Geometric Software (Survey)

**Epic / Feature** Geometric Software

**Business Value** Select, compile, and use trusted geometry libraries to accelerate development. (Geometric Software (Survey)).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master Geometric Software (Survey) so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Inventory CGAL/LEDA/libigl/VTK; record kernel differences and license terms.
- ☐ Build a minimal 'hello geometry' app creating a hull and a Delaunay triangulation.
- ☐ Decide on a primary stack for subsequent labs; justify trade-offs.

## HDCG-68 — LEDA & CGAL Case Studies

HDCG-68

LEDA & CGAL Case Studies

**Epic / Feature** Geometric Software

**Business Value** Select, compile, and use trusted geometry libraries to accelerate development. (LEDA & CGAL Case Studies).

**Priority / Estimate** **Priority:** Must

**SP:** 3

**Persona** applied geometry engineer

**Dependencies** prior chapters as referenced

**Assumptions / Risks** time to internalize proofs vs. breadth

*As a applied geometry engineer, I want to master LEDA & CGAL Case Studies so that I can apply it to real problems and communicate algorithmic insights clearly and reproducibly.*

**Non-Functional** Reliability Reproducibility

**Acceptance Criteria (BDD)**

**Scenario** Happy path

**Given** the chapter, examples, and any tooling are available

**When** I complete the *Hands-on Objective* and validations for this chapter

**Then** the stated outcomes are produced (proof/code/summary) and recorded in the repo with passing checks

**Definition of Ready:** Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed/flagged.

### Tasks

- ☐ Extract key definitions/lemmas; compile a one-page summary with references to the chapter.
- ☐ Compile and run sample kernels; implement hull→Delaunay→point-location pipeline.
- ☐ Add robust predicates and exact constructions where available.
- ☐ Package as a reusable module with unit tests and CLI demo.