Feasibility → satisfies constraints] → sub values of vars
Optimality → max/min value]

Graphical Method →
1. Define obj()
2. Draw constraints → Draw line for LHS = RHS
   ↳ Draw arrows
3. Find vertices of feasible region
4. Find value of obj @ vertices
5. ~~Pick the max/min~~ value

Primal Simplex Direct →
1. RHS +ve → else, multiply by -1
2. Convert inequality to equality
   add slack vars
3. Transpose obj()
4. Select

	entering	exiting
max	most -ve	least ratio
min	most +ve	least ratio

5. Calc pivot row
6. Check optimality → obj() coeff ≥ 0 ∀ non-basic vars
   ↳ verify feasibility
7. else calc other rows

den > 0

$$\text{Ratio} = \frac{\text{soln for basic var}}{\text{constraint coeff for entering var}}\ ; \quad \text{pivot row} = \frac{\text{exiting row}}{\text{pivot element}}$$

other rows = old row − (coeff in pivot col × pivot row)

→ if vars come one eqn each, treat them as basic vars
row operations to transform ~~constraint coeff~~ :
constraint coeff = 1
obj() coeff = 0

Artificial Starting Soln → Slack & artificial var tried for leaving
↳ Artificial leaves
↳ Surplus initially not basic
→ if any constraint has -ve RHS, × -1
↳ convert to equality
≤ ⟶ + Slack var
= ⟶ + Artificial var
≥ ⟶ − Surplus var + Artificial var

Summary	
All =	Solve
All ≤	Direct simplex
All ≥	Big M | 2 phase
Mix	"

Big M
↳ introduce artificial vars ; transposed obj()
M very large ; artificial anti-entering
↳ row transformation to eliminate R_1, R_2 in z row
↳ Solve as usual.

Max	Min
$+MR_1$ $+MR_2$	$-MR_1$ $-MR_2$
$z - MR_1 - MR_2$	$z + MR_1 + MR_2$

2 Phase Method
Phase 1 → R = min ΣR_i
↳ transform R_i as 0 in R row
↳ Solve as usual

Phase 2	artificial vars in basis		
R > 0	N/A	Infeasible X	—
R = 0	= 0	optimal soln of phase 2 LP	→ Drop col in phase 1 table corr to artificial vars → Use original obj() w/ constraints from phase 1 optimal table → phase 2 obj() → Row operation to eliminate phase 1 basic vars in phase 2 obj() row → Solve as usual
R = 0	≥ 0	optimal soln of original LP	Drop all → non-basic artificial vars from phase 1 table ↳ vars from org prob with -ve coeff in row of optimal phase 1 table Solve as usual

Special cases
Degeneracy → Redundant constraint → Soln for basic var = 0
Alt. optima → obj ∥ constraint → coeff of non-basic var = 0
Unbounded soln → entries for one/more non-basic var col ≤ 0 ∀ constraint rows
No feasible soln → artificial var exists in basis ; obj() row ≤/≥ max/min

Duality → constraint RHS & vars ≥ 0
→ ~~Add~~ Introduce slack | supplus (no artificial vars)
→ Dual var ∀ primal constraint
Dual constraint ∀ primal var (including slack | supplus)

primal obj()	dual obj()	inequality
max	min	≥
min	max	≤

Dual obj() = Σ RHS constraints × x_i
Dual constraint j = Σ constraint coeff of x_i ≤ inequality non-transposed obj() coeff of x_i

Formulae
→ [Primal constraint col in iteration i (primal basic vars)] = [inverse] × [primal constraints col]
→ primal obj() of x_j (primal non-basic vars) = LHS of non-transposed dual constraint − RHS " "
→ [Dual vars] = [RHS of basic vars in nontransposed dual constr] × [inverse]
→ [Inverse] = [columns in optimal primal table corr to vars not in obj()]

optimality & feasibility →

	max	min
feasibility	primal basic vars ≥ 0	←
optimality	primal non-basic vars ≥ 0	primal non-basic vars ≤ 0

Dual simplex

for	
≤	
≥	× 1
=	LHS ≤ RHS ; LHS ≥ RHS

2. Add slack vars (no supplus/artificial)
3. transpose obj()
4. Select entering/exiting vars (same for min/max)
exiting → most −ve
entering → least ratio
5. Calc pivot row
6. Check feasibility → value of basic vars ≥ 0 → verify optimality → end
7. Calc other rows

If constraint coeff ≥ 0 ∀ non-basic vars, no feasible soln.

Ratio = | obj() coeff / constraint coeff for exiting var | → magnitude, den < 0

Post Optimal Analysis

Change in feasibility

RHS of constraint changes	1. Check feasibility w/ inverse method 2. Calc new obj(), using values from step 1 3. update table 4. Use dual simplex 5. Calc obj() when feasibility is maintained/obtained
Addition of new constraint	1. Check feasibility by substituting existing values into new constraint 2. Introduce supplus/slack into eqn 3. Introduce supplus/slack ~~into~~ row & col into table 4. Update ~~all~~ introduced row using formula 5. Use dual simplex 6. Calc obj() when feasibility is maintained/obtained updated row of introduced basic var = initial row of introduced basic var − (Σ coeff i × x_i) coeff of x_i in new constraint; basic var in new constraint

Change in optimality
↳ due to change in obj()
1. Find dual var values, using new obj() coeff
2. Check optimality
3. If optimality maintained, go to step 6
4. update ~~obj~~ obj() row using
↳ coeff found when checking optimality
↳ original obj() value, using above latest coeff
5. Use primal simplex
6. Calculate latest soln. in new obj()

Summary

existing soln feasible?	existing soln optimal?	Action
✓	✓	No action
✓	✗	Primal simplex
✗	✓	Dual simplex
✗	✗	Generalized simplex

New constraint → Binding
→ redundant