MIT 6.00.2x — Quick Reference Cheat Sheet

RANDOM WALKS & STOCHASTIC PROCESSES

• Distance grows ~ sqrt(n)

• Run many trials for stable averages

• UsualDrunk: unbiased 4-direction walk

• ColdDrunk: biased walk (drift)

• Key pattern:

f = Field()

f.addDrunk(drunk, Location(0,0))

walk(f, drunk, steps)

SIMULATION & OOP ARCHITECTURE

Location:

Immutable (x,y); move returns new object

Field:

Maps drunks → locations

moveDrunk uses drunk.takeStep()

Drunk subclasses:

Define step behavior (polymorphism)

Robot hierarchy:

Robot → StandardRobot / RandomWalkRobot

Standard: keep direction until wall

RandomWalk: pick new direction each step

Simulation loop:

while coverage < target:

for r in robots: r.updatePositionAndClean()

OPTIMIZATION & DP (HIGH YIELD)

Brute Force:

Try all subsets → O(2^n)

Greedy:

Good heuristic, NOT optimal for 0/1 knapsack

Dynamic Programming:

Requires:

1. Optimal substructure

2. Overlapping subproblems

Use memo keys:

(len(items_left), remaining_capacity)

fastMaxVal (DP knapsack):

Top■down recursion + memoization

PLOTTING (PYLAB)

Core:

plt.plot(x,y)

plt.title("Title")

plt.xlabel("X")

plt.ylabel("Y")

plt.legend()

plt.show()

Histograms:

plt.hist(values, bins)

Multiple curves:

plt.plot(x,y1,label="A")

plt.plot(x,y2,label="B")

plt.legend()

PYTHON FUNDAMENTALS

Types:

int, float, str, bool

list (mutable), tuple (immutable)

dict, set

Functions:

def f(x): return x+1

Passed by object reference

Lists:

lst[i], lst[i:j], lst[::-1]

Dictionaries:

d[key] → value

d.items(), d.keys()

Strings:

Immutable

s.split(), s.join(), s.upper()

Exceptions:

try/except, raise

Common pitfalls:

• Mutable default args

• Modifying list while iterating

• Integer vs float division

END OF QUICK REFERENCE
