

Priority Queue Operations - heapq import heapq pq = [] heapq.heappush(pq, (2, "Harry")) heapq.heappush(pq, (3, "Charles")) heapq.heappush(pq, (1, "Riya")) heapq.heappush(pq, (4, "Stacy")) while pq: print(heapq.heappop(pq)) thread safe	Priority Queue Operations - PriorityQueue from queue import PriorityQueue pq = PriorityQueue() pq.put((2, "Harry")) pq.put((3, "Charles")) pq.put((1, "Riya")) pq.put((4, "Stacy")) while not pq.empty(): print(pq.get()) it is to implement thread
from queue import PriorityQueue # Simulated customer query customer_query = "I want to return my headphones" # Candidate responses with scores # Format: (urgency_score, purchase_match_score, browsing_match_score, response_text) responses = [(1, 0.9, 0.2, "You can start a return for your headphones here."), (3, 0.8, 0.5, "Looking for new headphones? Check these out."), (2, 0.7, 0.6, "Your order is on the way."), (1, 0.85, 0.3, "Refunds are processed within 5–7 business days.")] # Create a priority queue pq = PriorityQueue() # Insert responses with a composite priority tuple for urgency, purchase_score, browse_score, text in responses: # Lower urgency = higher priority, so we invert scores for max-heap behavior priority = (-urgency, -purchase_score, -browse_score) pq.put((priority, text)) # Get the top-ranked response top_response = pq.get()[1] print("💬 Chatbot Response:", top_response)	