

Angus has decided he needs to work primarily with *lines* rather than *characters*. He asks your opinion on the best way to proceed. Still flushed with the excitement of your new-found algorithmic knowledge from COSC262 you suggest using instead an *edit-distance* algorithm, using whole lines as the units of comparison rather than characters. You point out to Angus that this algorithm could distinguish between deleted lines, inserted lines, altered or substituted lines and unaltered lines, which will allow for a much nicer display. With great enthusiasm, Angus goes away for a bit and returns with a picture of the sort of display he now wants:

Previous	Current
# ===== DELETes =====	# ===== DELETes =====
# TODO: add docstrings	
@app.route('/queue/<hostname>', methods=['DELETE'])	@app.route('/queue/<hostname>', methods=['DELETE'])
def delete(hostname):	def delete(hostname):
	"""Handle delete request from the given host"""
try:	try:
data = json.loads(request.get_data())	data = json.loads(request.get_data())
mac_address = data['macAddress']	mac_address = data['mac_address']
except:	except:
abort(400, 'Missing or invalid user data')	abort(400, 'Missing or invalid user data')
status = queue.dequeue(hostname, macAddress)	status = queue.dequeue(hostname, mac_address)
return ('', status)	return ('', status)
@app.route('/queue', methods=['DELETE'])	@app.route('/queue', methods=['DELETE'])
	def clear_queue():
def empty_queue():	"""Clear the queue. Valid only if coming from tutor machine"""
if request.remote_addr.upper() != TUTOR_MACHINE.upper():	if request.remote_addr.upper() != TUTOR_MACHINE.upper():
abort(403, "Not authorised")	abort(403, "Only the tutor machine can clear the queue")
else:	else:
queue.clear_queue()	queue.clear_queue()
response = jsonify({"message": "Queue emptied"})	response = jsonify({"message": "Queue cleared"})
response.status_code = 204	response.status_code = 204
return response	return response

In this new display, deleted lines (which are always in the left column) are shown in purple, inserted lines (always in the right column) in green and altered or replaced lines in yellow.

Angus's delight with your new program lasts for several minutes. Then he says "You know, what would be really nice would be to highlight the actual modified characters within each changed line." He returns a bit later and adds "... like this for example".

Previous	Current
# ===== DELETes =====	# ===== DELETes =====
# TODO: add docstrings	
@app.route('/queue/<hostname>', methods=['DELETE'])	@app.route('/queue/<hostname>', methods=['DELETE'])
def delete(hostname):	def delete(hostname):
	"""Handle delete request from the given host"""
try:	try:
data = json.loads(request.get_data())	data = json.loads(request.get_data())
mac_address = data['macAddress']	mac_address = data['mac_address']
except:	except:
abort(400, 'Missing or invalid user data')	abort(400, 'Missing or invalid user data')
status = queue.dequeue(hostname, macAddress)	status = queue.dequeue(hostname, mac_address)
return ('', status)	return ('', status)
@app.route('/queue', methods=['DELETE'])	@app.route('/queue', methods=['DELETE'])
	def clear_queue():
def empty_queue():	"""Clear the queue. Valid only if coming from tutor machine"""
if request.remote_addr.upper() != TUTOR_MACHINE.upper():	if request.remote_addr.upper() != TUTOR_MACHINE.upper():
abort(403, "Not authorised")	abort(403, "Only the tutor machine can clear the queue")
else:	else:
queue.clear_queue()	queue.clear_queue()
response = jsonify({"message": "Queue emptied"})	response = jsonify({"message": "Queue cleared"})
response.status_code = 204	response.status_code = 204
return response	return response