



ΠΕΡΙΦΕΡΕΙΑ ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ 2014-2020
ΕΥΔ ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
Επενδυτικά Σχέδια Καινοτομίας
Κωδικός πράξης: KMP6-0083129
Κωδικός MIS: 5136571

Δικαιούχος: ΑΔΕΛΦΟΙ ΓΕΩΡΓΙΟΥ ΚΟΥΡΤΙΔΗ, ΕΛΚΕ – ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ

ΤΙΤΛΟΣ

«Βελτιστοποίηση τοποθέτησης και καταμέτρησης εμπορευμάτων σε μεγάλους βιομηχανικούς χώρους με χρήση μη επανδρωμένων αεροσκαφών»

Τίτλος Πράξης (Αγγλικά) Optimization of placement and counting products in large industrial areas using UAV

Παραδοτέο

Π3.4 Πρόγραμμα σε ελεύθερο λογισμικό βέλτιστης τοποθέτησης παραγόμενων βιομηχανικών προϊόντων

Αρ. Παραδοτέου	Π3.4
Ενότητα Εργασίας	ΕΕ3. Ανάπτυξη μεθόδων αυτόματης καταμέτρησης και βέλτιστης τοποθέτησης βιομηχανικών προϊόντων
Υπεύθυνος Φορέας ΕΕ3 / Π3	Αδελφοί Γεωργίου Κουρτίδη - ΔΙ.ΠΑ.Ε.
Είδος Παραδοτέου	Τεχνική Αναφορά
Παράδοση	Δεκέμβριος 2023
Σύντομη Περιγραφή Ενότητας	Πραγματοποίηση δοκιμαστικών πτήσεων για διαμόρφωση των ΣμηΕΑ και τη συλλογή δεδομένων, ανάπτυξη μεθόδων για την καταγραφή θέσης και ποσότητας προϊόντων, ανάπτυξη μεθόδων βέλτιστης τοποθέτησης προϊόντων, πρόγραμμα σε ελεύθερο λογισμικό.
Έναρξη Προγράμματος, Λήξη	Οκτώβριος 2021, Δεκέμβριος 2024
Φορείς	ΑΔΕΛΦΟΙ ΓΕΩΡΓΙΟΥ ΚΟΥΡΤΙΔΗ, ΕΛΚΕ – ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ



Λίστα συγγραφέων:

#	Όνοματεπώνυμο	Φορέας	email
1	Γεώργιος Κουρτίδης	ΝΕΚΤΑΡ	info@nektar.gr
2	Γεώργιος Κουρτίδης	ΝΕΚΤΑΡ	info@nektar.gr
3	Ειρήνη Κουρτίδου	ΝΕΚΤΑΡ	info@nektar.gr
4	Αναστάσιος Βαζικόγλου	ΝΕΚΤΑΡ	info@nektar.gr
5	Δημήτριος Βαρσάμης	ΔΙ.ΠΑ.Ε.	dvarsam@ihu.gr
6	Αλκιβιάδης Τσιμπίρης	ΔΙ.ΠΑ.Ε.	atsimpiris@ihu.gr
7	Κωνσταντίνος Χειλάς	ΔΙ.ΠΑ.Ε.	chilas@ihu.gr
8	Πάρις Μαστοροκώστας	ΔΙ.ΠΑ.Ε.	mast@uniwa.gr
9	Δημήτριος Μάνος	ΔΙ.ΠΑ.Ε.	dmanos@ihu.gr
10	Ιορδάνης Ζιώγας	ΔΙ.ΠΑ.Ε.	ziogasi@ihu.gr
11	Ηλίας Πανταζής	ΔΙ.ΠΑ.Ε.	hpant@ihu.gr

Επιμέλεια εγγράφου:

#	Όνοματεπώνυμο	Φορέας	email
	Γεώργιος Κουρτίδης	ΝΕΚΤΑΡ	info@nektar.gr
	Δημήτριος Βαρσάμης	ΔΙ.ΠΑ.Ε.	dvarsam@ihu.gr

Ιστορικό εγγράφου:

Έκδοση	Ημ/νια	Είδος εγγράφου	Φορέας
0.1		1 st draft	ΔΙ.ΠΑ.Ε.



Περιεχόμενα

1. Περίληψη έργου	4
2. Περίληψη Ενότητας Εργασίας 3 (ΕΕ3).....	5
3. Πρόγραμμα σε ελεύθερο λογισμικό (π.χ. Python) βέλτιστης τοποθέτησης παραγόμενων βιομηχανικών προϊόντων.....	6
4. Ανάλυση Κώδικα Python	11
Ρυθμίσεις και Παραμέτρους	Error! Bookmark not defined.
Βοηθητικές Συναρτήσεις	Error! Bookmark not defined.
☒ random_cell_value(): Τυχαία δημιουργία περιεχομένου κυψέλης....	Error! Bookmark not defined.
☒ manhattan_distance(): Υπολογισμός της "απόστασης Manhattan". .	Error! Bookmark not defined.
☒ print_grid(): Μορφοποιημένη εκτύπωση πινάκων.....	Error! Bookmark not defined.
☒ describe_path(): Γραφική περιγραφή διαδρομής μεταξύ δύο σημείων.	Error! Bookmark not defined.
☒ find_nearest_point(): Επιστροφή κοντινότερου σημείου εισόδου. ...	Error! Bookmark not defined.
☒ get_available_space() / get_total_stock(): Επιστροφή διαθέσιμου χώρου ή συνολικού αποθέματος ανά προϊόν.	Error! Bookmark not defined.
Δημιουργία Αποθήκης.....	Error! Bookmark not defined.
Τοποθέτηση Προϊόντων - allocate_products().....	Error! Bookmark not defined.
Παραλαβή Προϊόντων (Παραγγελία) - fulfill_order()	Error! Bookmark not defined.
Εκτέλεση Προγράμματος	Error! Bookmark not defined.
Συμπεράσματα	13
5. Παράδειγμα.....	14
6. Κατάλογος όρων και συντομογραφιών.....	19



1. Περίληψη έργου

Ο σκοπός του παρόντος έργου είναι η ανάπτυξη ενός ολοκληρωμένου πληροφοριακού συστήματος με στόχο την βέλτιστη τοποθέτηση εμπορευματοκιβωτίων σε μεγάλους βιομηχανικούς χώρους. Για την επίτευξη του αρχικά, θα δημιουργηθεί μία κεντρική σχεσιακή βάση δεδομένων η οποία θα ενοποιεί δεδομένα από διαφορετικές πηγές και χρονικές στιγμές, της επιχείρησης. Έπειτα, θα γίνει καθορισμός των απαιτήσεων και μεθόδων για την ανάλυση των δεδομένων της βάσης μέσω της σκιαγράφησης ενός γενικότερου προτύπου εκτέλεσης όλων των διαδικασιών παραγωγής. Η αποθήκη δεδομένων (warehouse), θα λειτουργήσει ως βάση για την ανάπτυξη και εφαρμογή αλγορίθμων που έχουν ως στόχο την εξαγωγή και παρουσίαση Business Analytics υπό μορφή γραφημάτων, πινάκων και δεικτών.

Ακολούθως, θα πραγματοποιηθούν δοκιμαστικές πτήσεις των ειδικών για το έργο ΣμηΕΑ, πρωτίστως για την διαμόρφωσή τους και εν συνεχεία για την καταγραφή της θέσης και της διαθέσιμης ποσότητας του κάθε προϊόντος. Με βάση τα δεδομένα αυτά, θα αναπτυχθούν υπολογιστικές μέθοδοι οι οποίες θα είναι σε θέση να εντοπίζουν το ζητούμενο προϊόν και να υπολογίζουν την διαθεσιμότητά του. Σύμφωνα με τα δεδομένα αυτά, αλλά και με χρήση business analytics, προκύπτουν συμπεράσματα σχετικά με την ζήτηση και την αναγκαία επάρκεια των προϊόντων.

Στη συνέχεια θα αναπτυχθεί μία ειδική μέθοδος βελτιστοποίησης η οποία με δεδομένα όλα τα παραπάνω στοχεύει στην βέλτιστη τοποθέτηση των παραγόμενων προϊόντων στην αποθήκη με βάση συγκεκριμένες προδιαγραφές.

Τέλος, θα αναπτυχθεί ολοκληρωμένο πληροφοριακό σύστημα το οποίο θα εκτελεί αυτόματη καταμέτρηση της αποθήκης και θα προτείνει την βέλτιστη τοποθέτηση προϊόντων βάση προδιαγραφών που θα δίνονται από τον χρήστη. Το σύστημα αυτό, γίνεται πράξη μέσω της ανάπτυξης ολοκληρωμένης πλατφόρμας αλλά και εφαρμογής για έξυπνες κινητές συσκευές, με τη βοήθεια της οποίας οι εργαζόμενοι της επιχείρησης θα μπορούν να έχουν εικόνα της αποθήκης προϊόντων κάθε στιγμή καθώς και την δυνατότητα μετάδοσης της εικόνας αυτής μεταξύ συνεργατών μέσω ηλεκτρονικού ταχυδρομείου ή κινητής τηλεφωνίας για την επίτευξη της καλύτερης δυνατής επικοινωνίας και συνεργασίας αυτών.

Το έργο λοιπόν, υποβοηθώντας στη διαδικασία λήψης αποφάσεων, θα συνδράμει σημαντικά στη μείωση των εξόδων της εταιρίας, θα μειώσει τον φόρτο εργασίας των εργαζομένων απαλλάσσοντάς τους από τετριμμένες διαδικασίες και θα καταστήσει την επιχείρηση ιδιαίτερα ανταγωνιστική στον χώρο της παραγωγής. Επιπρόσθετα, η αξία ενός τέτοιου προϊόντος δεν περιορίζεται μόνο στο λειτουργικό του κομμάτι αλλά εστιάζεται και στο εμπορικό, καθιστώντας την επιχείρηση που το διαθέτει άκρως ανταγωνιστική στην αγορά εργασίας, καθώς μία τέτοια εφαρμογή προσδίδει υπεραξία στην επιχείρηση αν επιλεγεί η εμπορική της εκμετάλλευσή της.

Το έργο αποτελείται από έξι επιμέρους ενότητες εργασίας, η ολοκλήρωση των οποίων θα οδηγήσει στην επίτευξη του έργου. Οι ενότητες αυτές είναι:

- EE1 Σχεδίαση υλοποίησης έργου
- EE2 Ανάπτυξη μεθόδων και συστημάτων οργάνωσης δεδομένων



- ΕΕ3 Ανάπτυξη μεθόδων αυτόματης καταμέτρησης και βέλτιστης τοποθέτησης βιομηχανικών προϊόντων
- ΕΕ4 Ανάπτυξη διαδικτυακής πλατφόρμας και εφαρμογής σε έξυπνες κινητές συσκευές
- ΕΕ5 Πιλοτική λειτουργία και αξιολόγηση ολοκληρωμένου συστήματος
- ΕΕ6 Υποστήριξη έργου

2. Περίληψη Ενότητας Εργασίας 3 (ΕΕ3)

Κατά τη διάρκεια της ΕΕ3 θα πραγματοποιηθούν δοκιμαστικές πτήσεις για την βέλτιστη διαμόρφωση των ΣμηΕΑ. Παράλληλα, με βάση τα δεδομένα που θα συλλεχθούν, θα αναζητηθούν μέθοδοι και παραμετροποιήσεις έτσι ώστε να πραγματοποιείται σωστή καταγραφή των προϊόντων ως προς την θέση τους και ως προς την ποσότητα τους. Θα αναπτυχθούν εξειδικευμένες υπολογιστικές μέθοδοι οι οποίες με βάση τα δεδομένα που θα συλλέγουν τα ΣμηΕΑ, θα εντοπίζουν το προϊόν, τη θέση του και θα καταμετρούν την ποσότητα ανά θέση και ανά προϊόν. Με βάση τις πληροφορίες από την αποθήκη δεδομένων και τα αποτελέσματα των υπολογιστικών μεθόδων, θα αναπτυχθεί μια ειδική μέθοδος βελτιστοποίησης η οποία έχει σκοπό τη βέλτιστη τοποθέτηση των παραγόμενων προϊόντων στην αποθήκη με βάση συγκεκριμένες προδιαγραφές, για παράδειγμα με βάση τις αναμενόμενες πωλήσεις, τους διαθέσιμους πόρους κ.α.

Παραδοτέα:

- Π3.1 Μέθοδος αυτόματης καταμέτρησης βιομηχανικών προϊόντων
- Π3.2 Μέθοδος βέλτιστης τοποθέτησης παραγόμενων βιομηχανικών προϊόντων
- Π3.3 Πρόγραμμα σε ελεύθερο λογισμικό (π.χ. Python) αυτόματης καταμέτρησης βιομηχανικών προϊόντων
- Π3.4 Πρόγραμμα σε ελεύθερο λογισμικό (π.χ. Python) βέλτιστης τοποθέτησης παραγόμενων βιομηχανικών προϊόντων



3. Πρόγραμμα σε ελεύθερο λογισμικό (π.χ. Python) βέλτιστης τοποθέτησης παραγόμενων βιομηχανικών προϊόντων

Παρακάτω αναλυτικά ο κώδικας σε γλώσσα Python, για τη βέλτιστη τοποθέτηση των παραγόμενων βιομηχανικών προϊόντων:

```
import random
from collections import defaultdict
import copy

# --- Ρυθμίσεις ---
letter_limits = {'A': 5, 'B': 5, 'C': 6, 'D': 7}
rows, cols = 10, 13
entry_points = [(0, 6), (9, 6)]
symbol_map = {(0, 6): '↓', (9, 6): '↑'}

# --- Βοηθητικές ---
def random_cell_value():
    letter = random.choice(list(letter_limits.keys()))
    value = random.randint(0, letter_limits[letter])
    return f"{letter}{value}", str(letter_limits[letter] - value)

def manhattan_distance(x1, y1, x2, y2):
    return abs(x1 - x2) + abs(y1 - y2)

def print_grid(grid, title):
    print(f"\n{title}")
    for row in grid:
        print(' '.join(f"{cell:>3}" for cell in row))

def describe_path(start, target):
    path = []
    r0, c0 = start
    r, c = target
    while r0 < r: path.append('↓'); r0 += 1
    while r0 > r: path.append('↑'); r0 -= 1
    while c0 < c: path.append('→'); c0 += 1
```



```
while c0 > c: path.append('←'); c0 -= 1
return ''.join(path)

def find_nearest_point(cell):
    return min(entry_points, key=lambda p: manhattan_distance(p[0], p[1],
cell[0], cell[1]))

def get_available_space(warehouse):
    available = defaultdict(int)
    for r in range(rows):
        for c in range(cols):
            cell = warehouse[r][c]
            if isinstance(cell, str) and len(cell) == 2:
                letter = cell[0]
                val = int(cell[1])
                available[letter] += letter_limits[letter] - val
    return available

def get_total_stock(warehouse):
    stock = defaultdict(int)
    for r in range(rows):
        for c in range(cols):
            cell = warehouse[r][c]
            if isinstance(cell, str) and len(cell) == 2:
                letter = cell[0]
                val = int(cell[1])
                stock[letter] += val # Σωστό: προσθέτουμε την ποσότητα, όχι το
πλήθος παλετιών
    return stock

# --- Δημιουργία αποθήκης ---
def create_warehouse():
    warehouse, capacity = [], []
    for r in range(rows):
        w_row, c_row = [], []
        for c in range(cols):
            if r in [0, 3, 6, 9] or c == 6:
                w_row.append('*')
```



```
        c_row.append(' ')
    else:
        val, rem = random_cell_value()
        w_row.append(val)
        c_row.append(rem)
    warehouse.append(w_row)
    capacity.append(c_row)
for ep in entry_points:
    warehouse[ep[0]][ep[1]] = symbol_map[ep]
    capacity[ep[0]][ep[1]] = ' '
return warehouse, capacity

# --- Τοποθέτηση προϊόντων ---
def allocate_products(warehouse, capacity, quantities):
    candidates = defaultdict(list)
    allocations = []
    total_available = get_available_space(warehouse)

    for r in range(rows):
        for c in range(cols):
            cell = warehouse[r][c]
            if isinstance(cell, str) and len(cell) == 2:
                letter = cell[0]
                value = int(cell[1])
                avail = letter_limits[letter] - value
                if avail > 0:
                    nearest = find_nearest_point((r, c))
                    dist = manhattan_distance(nearest[0], nearest[1], r, c)
                    candidates[letter].append({'row': r, 'col': c, 'available':
avail, 'dist': dist, 'from': nearest})

    for letter, qty in quantities.items():
        if qty > total_available[letter]:
            print(f"✘ Δεν υπάρχει αρκετός διαθέσιμος χώρος για {letter} (ζητήθηκε
{qty}, διαθέσιμο {total_available[letter]})")
            return []

    for letter in candidates:
```



```
    candidates[letter].sort(key=lambda x: x['dist'])

for letter, qty in quantities.items():
    for cell in candidates[letter]:
        if qty <= 0:
            break
        put = min(qty, cell['available'])
        qty -= put
        allocations.append({'letter': letter, 'row': cell['row'], 'col':
cell['col'], 'placed': put, 'from': cell['from']})
        current_val = int(warehouse[cell['row']][cell['col']][1])
        new_val = current_val + put
        warehouse[cell['row']][cell['col']] = f"{letter}{new_val}"
        capacity[cell['row']][cell['col']] = str(letter_limits[letter] -
new_val)
    return allocations

# --- Παραλαβή (παραγγελία) ---
def fulfill_order(warehouse, capacity, quantities):
    candidates = defaultdict(list)
    extractions = []
    total_stock = get_total_stock(warehouse)

    for r in range(rows):
        for c in range(cols):
            cell = warehouse[r][c]
            if isinstance(cell, str) and len(cell) == 2:
                letter = cell[0]
                val = int(cell[1])
                if val > 0:
                    nearest = find_nearest_point((r, c))
                    dist = manhattan_distance(nearest[0], nearest[1], r, c)
                    candidates[letter].append({'row': r, 'col': c, 'value': val,
'dist': dist, 'to': nearest})

    for letter, qty in quantities.items():
        if qty > total_stock[letter]:
            print(f"✘ Δεν υπάρχει αρκετό απόθεμα για {letter} (ζητήθηκε {qty},
διαθέσιμο {total_stock[letter]})")
```



```
        return []

    for letter in candidates:
        candidates[letter].sort(key=lambda x: x['dist'])

    for letter, qty in quantities.items():
        for cell in candidates[letter]:
            if qty <= 0:
                break
            take = min(qty, cell['value'])
            qty -= take
            extractions.append({'letter': letter, 'row': cell['row'], 'col':
cell['col'], 'taken': take, 'to': cell['to']})
            current_val = int(warehouse[cell['row']][cell['col']][1])
            new_val = current_val - take
            warehouse[cell['row']][cell['col']] = f"{letter}{new_val}"
            capacity[cell['row']][cell['col']] = str(letter_limits[letter] -
new_val)
        return extractions

# --- Εκτέλεση ---
warehouse, capacity = create_warehouse()
initial_warehouse = copy.deepcopy(warehouse)
initial_capacity = copy.deepcopy(capacity)

print_grid(initial_warehouse, "\n📦 Αρχική Αποθήκη")
print_grid(initial_capacity, "📦 Αρχικά Υπολειπόμενα")

available = get_available_space(warehouse)
input_quantities = {}
for l in 'ABCD':
    print(f"Διαθέσιμος χώρος για {l}: {available[l]}")
    input_quantities[l] = int(input(f"Ποσότητα για τοποθέτηση {l}: "))

allocs = allocate_products(warehouse, capacity, input_quantities)
if allocs:
    print("\n✅ Τοποθετήσεις:")
    for a in allocs:
```



```
path = describe_path(a['from'], (a['row'], a['col']))
print(f"{a['letter']} → ({a['row']},{a['col']}), ποσότητα: {a['placed']},
από: {a['from']}, διαδρομή: {path}")

print_grid(warehouse, "\n📦 Αποθήκη μετά την τοποθέτηση")
print_grid(capacity, "📦 Υπολειπόμενα μετά την τοποθέτηση")

stock = get_total_stock(warehouse)
order_quantities = {}
for l in 'ABCD':
    print(f"Διαθέσιμο απόθεμα για {l}: {stock[l]}")
    order_quantities[l] = int(input(f"Ποσότητα για παραγγελία {l}: "))

extractions = fulfill_order(warehouse, capacity, order_quantities)
if extractions:
    print("\n📦 Εκτελεσμένες Παραγγελίες:")
    for e in extractions:
        path = describe_path((e['row'], e['col']), e['to'])
        print(f"{e['letter']} ← ({e['row']},{e['col']}), ποσότητα:
{e['taken']}, προς: {e['to']}, διαδρομή: {path}")

print_grid(warehouse, "\n📦 Τελική Αποθήκη")
print_grid(capacity, "📦 Τελικά Υπολειπόμενα")
```

4. Ανάλυση Κώδικα Python

Ρυθμίσεις και Βοηθητικές Συναρτήσεις

- Ορίζονται:
 - Μέγιστη χωρητικότητα ανά τύπο προϊόντος (`letter_limits`)
 - Διαστάσεις της αποθήκης
 - Σημεία εισόδου/εξόδου (`entry_points`), με συμβολισμό ↕
- **random_cell_value**: Επιστρέφει τυχαίο προϊόν και διαθέσιμη χωρητικότητα.
- **manhattan_distance**: Υπολογίζει την απόσταση μεταξύ δύο σημείων στο πλέγμα.
- **print_grid**: Εκτυπώνει αναπαράσταση αποθήκης ή υπολοίπων.
- **describe_path**: Παράγει συμβολοσειρά διαδρομής (π.χ. "↓↓←→") μεταξύ σημείων.



- **find_nearest_point**: Βρίσκει το πλησιέστερο σημείο εισόδου σε ένα κελί.
- **get_available_space/get_total_stock**: Υπολογίζουν συνολικό διαθέσιμο χώρο ή συνολικό απόθεμα ανά προϊόν.

Δημιουργία Αποθήκης

def create_warehouse():

- Κάθε κελί παίρνει μια τιμή τύπου "Α3" (είδος και ποσότητα).
- Σε καθορισμένες γραμμές (0,3,6,9) και στήλη (6) τοποθετούνται εμπόδια (*).
- Τα σημεία εισόδου τοποθετούνται με ειδικό σύμβολο ↕.
- Η συνάρτηση επιστρέφει:
 - Τον πίνακα της αποθήκης (warehouse)
 - Τον πίνακα υπολειπόμενης χωρητικότητας (capacity)

Τοποθέτηση Προϊόντων

def allocate_products(warehouse, capacity, quantities):

Η διαδικασία δέχεται ως είσοδο τις ποσότητες προς τοποθέτηση και:

- Φιλτράρει κυψέλες με διαθέσιμο χώρο για κάθε προϊόν.
- Υπολογίζει την απόσταση κάθε κυψέλης από το πλησιέστερο σημείο εισόδου.
- Ταξινομεί τις θέσεις από μικρότερη σε μεγαλύτερη απόσταση.
- Τοποθετεί προϊόντα τμηματικά μέχρι να καλυφθεί η ποσότητα.
- Ενημερώνει τους πίνακες warehouse και capacity με τις νέες τιμές.

Αν δεν επαρκεί ο διαθέσιμος χώρος, εμφανίζεται μήνυμα σφάλματος.

Παραλαβή Προϊόντων (Εκτέλεση Παραγγελίας)

def fulfill_order(warehouse, capacity, quantities):

Η συνάρτηση δέχεται ως είσοδο τις ζητούμενες ποσότητες παραγγελίας και:

- Αναζητεί κυψέλες με διαθέσιμο απόθεμα.
- Υπολογίζει την απόσταση από κάθε κυψέλη στο πλησιέστερο σημείο εξόδου.
- Επιλέγει πρώτα τις πλησιέστερες θέσεις.
- Εξάγει τμηματικά το απόθεμα και ενημερώνει τους πίνακες.

Αν το συνολικό απόθεμα είναι ανεπαρκές, εμφανίζεται προειδοποίηση.



Εκτέλεση Προγράμματος

Η κύρια ροή του προγράμματος περιλαμβάνει:

- Δημιουργία αρχικής αποθήκης.
- Εμφάνιση κατάστασης.
- Εισαγωγή ποσοτήτων προς τοποθέτηση από χρήστη.
- Εκτέλεση τοποθέτησης και εμφάνιση διαδρομών.
- Εισαγωγή ποσοτήτων παραγγελίας.
- Εκτέλεση παραλαβών και τελική ενημέρωση.

Παρατηρήσεις και Δυνατότητες Επέκτασης

- Η χρήση **απόστασης Manhattan** βελτιστοποιεί τη διακίνηση.
- Τα αποτελέσματα αναπαρίστανται με σαφήνεια σε πίνακες.
- Ο χρήστης μπορεί να πειραματιστεί με σενάρια λειτουργίας.

Η προσέγγιση αυτή μπορεί να επεκταθεί ώστε να περιλαμβάνει:

- Οπτικοποίηση της αποθήκης με GUI ή Web Interface.
- Υποστήριξη προτεραιοτήτων (π.χ. προϊόντα με υψηλή ζήτηση).
- Συνδυασμό με αισθητήρες πραγματικού χρόνου για αυτοματισμό.

Συμπεράσματα

Ο κώδικας μοντελοποιεί με απλό και επεκτάσιμο τρόπο τις βασικές διαδικασίες **τοποθέτησης και παραλαβής προϊόντων** σε μία πλέγματοειδή αποθήκη. Ενσωματώνει αρχές βελτιστοποίησης με βάση απόσταση (Manhattan), ελέγχει περιορισμούς (όρια κυψέλης) και παρέχει οπτική αναπαράσταση. Προσφέρεται για μελλοντικές επεκτάσεις όπως:

- χρήση προτεραιοτήτων,
- real-time ενημέρωση,
- τρισδιάστατη απεικόνιση.

Πρόκειται για ισχυρό εργαλείο προσομοίωσης σε περιβάλλοντα αποθηκών ή logistics.



5. Παράδειγμα

Στο παράδειγμα που ακολουθεί έχουμε λάβει την καταμέτρηση (Αρχική Αποθήκη) καταχωρούμε την πρόβλεψη και τοποθετούνται με βέλτιστο τρόπο τα προϊόντα και με την παραγγελία εξάγουμε τα προϊόντα από την αποθήκη με βέλτιστο τρόπο.

Αρχική Αποθήκη

*	*	*	*	*	*	↓	*	*	*	*	*	*
B4	C0	D0	C2	C4	D7	*	D2	B0	A2	B2	D3	C6
B3	D6	B3	B2	C4	D3	*	B1	A0	D0	C6	A3	C1
*	*	*	*	*	*	*	*	*	*	*	*	*
B5	A4	C6	A4	A3	C4	*	D5	B3	D3	B3	D0	D4
D0	C5	A0	C3	A3	B5	*	B4	C3	A0	A4	D6	B4
*	*	*	*	*	*	*	*	*	*	*	*	*
C3	A4	C5	B5	B2	C2	*	A2	D7	C0	D3	B2	A0
D6	A1	B0	D1	C4	C6	*	B2	A0	C6	C5	C5	C4
*	*	*	*	*	*	↑	*	*	*	*	*	*

Αρχικά Υπολειπόμενα

1	6	7	4	2	0	5	5	3	3	4	0
2	1	2	3	2	4	4	5	7	0	2	5
0	1	0	1	2	2	2	2	4	2	7	3
7	1	5	3	2	0	1	3	5	1	1	1
3	1	1	0	3	4	3	0	6	4	3	5
1	4	5	6	2	0	3	5	0	1	1	2

Διαθέσιμος χώρος για A: 45

Ποσότητα για τοποθέτηση A: 40

Διαθέσιμος χώρος για B: 40

Ποσότητα για τοποθέτηση B: 40

Διαθέσιμος χώρος για C: 48



Ποσότητα για τοποθέτηση C: 40

Διαθέσιμος χώρος για D: 63

Ποσότητα για τοποθέτηση D: 40

Τοποθετήσεις:

A → (7,7), ποσότητα: 3, από: (9, 6), διαδρομή: ↑↑→
A → (8,8), ποσότητα: 5, από: (9, 6), διαδρομή: ↑→→
A → (1,9), ποσότητα: 3, από: (0, 6), διαδρομή: ↓→→→
A → (2,8), ποσότητα: 5, από: (0, 6), διαδρομή: ↓↓→→
A → (4,4), ποσότητα: 2, από: (0, 6), διαδρομή: ↓↓↓↓←←
A → (5,4), ποσότητα: 2, από: (9, 6), διαδρομή: ↑↑↑↑←←
A → (8,1), ποσότητα: 4, από: (9, 6), διαδρομή: ↑←←←←←
A → (2,11), ποσότητα: 2, από: (0, 6), διαδρομή: ↓↓→→→→→
A → (4,3), ποσότητα: 1, από: (0, 6), διαδρομή: ↓↓↓↓←←←←
A → (5,9), ποσότητα: 5, από: (9, 6), διαδρομή: ↑↑↑↑→→→
A → (7,1), ποσότητα: 1, από: (9, 6), διαδρομή: ↑↑←←←←←←
A → (5,2), ποσότητα: 5, από: (9, 6), διαδρομή: ↑↑↑↑←←←←←←
A → (5,10), ποσότητα: 1, από: (9, 6), διαδρομή: ↑↑↑↑→→→→→
A → (7,12), ποσότητα: 1, από: (9, 6), διαδρομή: ↑↑→→→→→→→
B → (8,7), ποσότητα: 3, από: (9, 6), διαδρομή: ↑→
B → (1,8), ποσότητα: 5, από: (0, 6), διαδρομή: ↓→→
B → (2,7), ποσότητα: 4, από: (0, 6), διαδρομή: ↓↓→
B → (7,4), ποσότητα: 3, από: (9, 6), διαδρομή: ↑↑←←
B → (1,10), ποσότητα: 3, από: (0, 6), διαδρομή: ↓→→→→
B → (2,3), ποσότητα: 3, από: (0, 6), διαδρομή: ↓↓←←←←
B → (5,7), ποσότητα: 1, από: (9, 6), διαδρομή: ↑↑↑↑→
B → (8,2), ποσότητα: 5, από: (9, 6), διαδρομή: ↑←←←←←
B → (2,2), ποσότητα: 2, από: (0, 6), διαδρομή: ↓↓←←←←←←
B → (4,8), ποσότητα: 2, από: (0, 6), διαδρομή: ↓↓↓↓→→
B → (1,0), ποσότητα: 1, από: (0, 6), διαδρομή: ↓←←←←←←←
B → (7,11), ποσότητα: 3, από: (9, 6), διαδρομή: ↑↑→→→→→
B → (2,0), ποσότητα: 2, από: (0, 6), διαδρομή: ↓↓←←←←←←
B → (4,10), ποσότητα: 2, από: (0, 6), διαδρομή: ↓↓↓↓→→→→→
B → (5,12), ποσότητα: 1, από: (9, 6), διαδρομή: ↑↑↑↑→→→→→→
C → (1,4), ποσότητα: 2, από: (0, 6), διαδρομή: ↓←←



C → (7,5), ποσότητα: 4, από: (9, 6), διαδρομή: ↑↑←
 C → (8,4), ποσότητα: 2, από: (9, 6), διαδρομή: ↑←←
 C → (1,3), ποσότητα: 4, από: (0, 6), διαδρομή: ↓←←←
 C → (2,4), ποσότητα: 2, από: (0, 6), διαδρομή: ↓↓←←
 C → (4,5), ποσότητα: 2, από: (0, 6), διαδρομή: ↓↓↓↓←
 C → (7,9), ποσότητα: 6, από: (9, 6), διαδρομή: ↑↑→→→
 C → (8,10), ποσότητα: 1, από: (9, 6), διαδρομή: ↑→→→→
 C → (1,1), ποσότητα: 6, από: (0, 6), διαδρομή: ↓←←←←
 C → (5,8), ποσότητα: 3, από: (9, 6), διαδρομή: ↑↑↑↑→
 C → (7,2), ποσότητα: 1, από: (9, 6), διαδρομή: ↑↑←←←
 C → (8,11), ποσότητα: 1, από: (9, 6), διαδρομή: ↑→→→→
 C → (5,3), ποσότητα: 3, από: (9, 6), διαδρομή: ↑↑↑↑←←
 C → (8,12), ποσότητα: 2, από: (9, 6), διαδρομή: ↑→→→→→
 C → (2,12), ποσότητα: 1, από: (0, 6), διαδρομή: ↓↓→→→→→
 D → (1,7), ποσότητα: 5, από: (0, 6), διαδρομή: ↓→
 D → (2,5), ποσότητα: 4, από: (0, 6), διαδρομή: ↓↓←
 D → (8,3), ποσότητα: 6, από: (9, 6), διαδρομή: ↑←←←
 D → (1,2), ποσότητα: 7, από: (0, 6), διαδρομή: ↓←←←←
 D → (2,9), ποσότητα: 7, από: (0, 6), διαδρομή: ↓↓→→→
 D → (4,7), ποσότητα: 2, από: (0, 6), διαδρομή: ↓↓↓↓→
 D → (1,11), ποσότητα: 4, από: (0, 6), διαδρομή: ↓→→→→
 D → (7,10), ποσότητα: 4, από: (9, 6), διαδρομή: ↑↑→→→
 D → (2,1), ποσότητα: 1, από: (0, 6), διαδρομή: ↓↓←←←←

Αποθήκη μετά την τοποθέτηση

*	*	*	*	*	*	↑	*	*	*	*	*	*
B5	C6	D7	C6	C6	D7	*	D7	B5	A5	B5	D7	C6
B5	D7	B5	B5	C6	D7	*	B5	A5	D7	C6	A5	C2
*	*	*	*	*	*	*	*	*	*	*	*	*
B5	A4	C6	A5	A5	C6	*	D7	B5	D3	B5	D0	D4
D0	C5	A5	C6	A5	B5	*	B5	C6	A5	A5	D6	B5
*	*	*	*	*	*	*	*	*	*	*	*	*
C3	A5	C6	B5	B5	C6	*	A5	D7	C6	D7	B5	A1
D6	A5	B5	D7	C6	C6	*	B5	A5	C6	C6	C6	C6



* * * * * † * * * *

Υπολειπόμενα μετά την τοποθέτηση

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	4
0	1	0	0	0	0	0	0	4	0	7	3
7	1	0	0	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	0	0	0	0	4
1	0	0	0	0	0	0	0	0	0	0	0

Διαθέσιμο απόθεμα για A: 70
 Ποσότητα για παραγγελία A: 40
 Διαθέσιμο απόθεμα για B: 90
 Ποσότητα για παραγγελία B: 40
 Διαθέσιμο απόθεμα για C: 124
 Ποσότητα για παραγγελία C: 40
 Διαθέσιμο απόθεμα για D: 96
 Ποσότητα για παραγγελία D: 40

Εκτελεσμένες Παραγγελίες:

A ← (7,7), ποσότητα: 5, προς: (9, 6), διαδρομή: ↓↓←
 A ← (8,8), ποσότητα: 5, προς: (9, 6), διαδρομή: ↓←←
 A ← (1,9), ποσότητα: 5, προς: (0, 6), διαδρομή: ↑←←←
 A ← (2,8), ποσότητα: 5, προς: (0, 6), διαδρομή: ↑↑←←
 A ← (4,4), ποσότητα: 5, προς: (0, 6), διαδρομή: ↑↑↑↑→
 A ← (5,4), ποσότητα: 5, προς: (9, 6), διαδρομή: ↓↓↓↓→
 A ← (8,1), ποσότητα: 5, προς: (9, 6), διαδρομή: ↓→→→→
 A ← (2,11), ποσότητα: 5, προς: (0, 6), διαδρομή: ↑↑←←←←
 B ← (8,7), ποσότητα: 5, προς: (9, 6), διαδρομή: ↓←
 B ← (1,8), ποσότητα: 5, προς: (0, 6), διαδρομή: ↑←←
 B ← (2,7), ποσότητα: 5, προς: (0, 6), διαδρομή: ↑↑←
 B ← (7,4), ποσότητα: 5, προς: (9, 6), διαδρομή: ↓↓→



B ← (1,10), ποσότητα: 5, προς: (0, 6), διαδρομή: ↑←←←←
 B ← (2,3), ποσότητα: 5, προς: (0, 6), διαδρομή: ↑↑→→→
 B ← (5,5), ποσότητα: 5, προς: (9, 6), διαδρομή: ↓↓↓↓→
 B ← (5,7), ποσότητα: 5, προς: (9, 6), διαδρομή: ↓↓↓↓←
 C ← (8,5), ποσότητα: 6, προς: (9, 6), διαδρομή: ↓→
 C ← (1,4), ποσότητα: 6, προς: (0, 6), διαδρομή: ↑→→
 C ← (7,5), ποσότητα: 6, προς: (9, 6), διαδρομή: ↓↓→
 C ← (8,4), ποσότητα: 6, προς: (9, 6), διαδρομή: ↓→→
 C ← (1,3), ποσότητα: 6, προς: (0, 6), διαδρομή: ↑→→→
 C ← (2,4), ποσότητα: 6, προς: (0, 6), διαδρομή: ↑↑→→
 C ← (8,9), ποσότητα: 4, προς: (9, 6), διαδρομή: ↓←←←
 D ← (1,5), ποσότητα: 7, προς: (0, 6), διαδρομή: ↑→
 D ← (1,7), ποσότητα: 7, προς: (0, 6), διαδρομή: ↑←
 D ← (2,5), ποσότητα: 7, προς: (0, 6), διαδρομή: ↑↑→
 D ← (7,8), ποσότητα: 7, προς: (9, 6), διαδρομή: ↓↓←←
 D ← (8,3), ποσότητα: 7, προς: (9, 6), διαδρομή: ↓→→→
 D ← (1,2), ποσότητα: 5, προς: (0, 6), διαδρομή: ↑→→→→

Τελική Αποθήκη

*	*	*	*	*	*	↑	*	*	*	*	*	*
B5	C6	D2	C0	C0	D0	*	D0	B0	A0	B0	D7	C6
B5	D7	B5	B0	C0	D0	*	B0	A0	D7	C6	A0	C2
*	*	*	*	*	*	*	*	*	*	*	*	*
B5	A4	C6	A5	A0	C6	*	D7	B5	D3	B5	D0	D4
D0	C5	A5	C6	A0	B0	*	B0	C6	A5	A5	D6	B5
*	*	*	*	*	*	*	*	*	*	*	*	*
C3	A5	C6	B5	B0	C0	*	A0	D0	C6	D7	B5	A1
D6	A0	B5	D0	C0	C0	*	B0	A0	C2	C6	C6	C6
*	*	*	*	*	*	↑	*	*	*	*	*	*

Τελικά Υπολειπόμενα

0	0	5	6	6	7	7	5	5	5	0	0
0	0	0	5	6	7	5	5	0	0	5	4



0	1	0	0	5	0	0	0	4	0	7	3
7	1	0	0	5	5	5	0	0	0	1	0
3	0	0	0	5	6	5	7	0	0	0	4
1	5	0	7	6	6	5	5	4	0	0	0

6. Κατάλογος όρων και συντομογραφιών

ΣμηΕΑ	Συστήματα μη Επανδρωμένων Αεροσκαφών
Δι.ΠΑ.Ε.	Διεθνές Πανεπιστήμιο της Ελλάδος
ΠΑ.Δ.Α.	Πανεπιστήμιο Δυτικής Αττικής
Δ.Ε.Π.	Διδακτικό Ερευνητικό Προσωπικό
Ε.Δι.Π.	Εργαστηριακό Διδακτικό Προσωπικό
Α.Ε.Ι.	Ανώτατο Εκπαιδευτικό Ίδρυμα
Τ.Ε.Ι.	Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δ.Ο.Α.Τ.Α.Π.	Διεπιστημονικός Οργανισμός Αναγνώρισης Τίτλων Ακαδημαϊκών και πληροφόρησης