

Αναπαράσταση του κόσμου του προβλήματος του racman με Python

Το συγκεκριμένο Πρόγραμμα είναι η αναπαράσταση του εσωτερικού κόσμου το rasman. Στην αρχή ο εσωτερικός κόσμος αποτελείτε από τα εξής κομμάτια:

- Το πρώτο κομμάτι της αναπαράστασης είναι οι καταστάσεις αρχής και τέλους. Η αναπαράσταση γίνεται με λίστα όπου για στοιχεία έχει λίστες όπου αντιπροσωπεύουν τα κελιά του κόσμου του `pacman`.

[[",","],[",","],[",","],[",","],[",","],[",","],[",",']]]

- Και η αναπαράσταση των στοιχείων που βρίσκονται στην λίστα είναι
 - ο `pacman` (`p`) : [`'p'`,`''`]
 - τα φρούτα (`a,o`) : [`'p'`,`'a'`] [`'p'`,`'o'`]
 - το `block` (`b`) : [`'p'`,`'b'`]

Το συγκεκριμένο πρόβλημα είναι ότι ο rasman πρέπει να φάει όλα τα επειτρεπτά φρούτα και να σπάσει όλα τα μπλοκ που βρίσκει μπροστά του και να καταλήξει στο τελευταίο κελί του κόσμου .

Οι επηρεαζόμενες κινήσεις είναι :

- Ο pacman πηγαίνει μόνο δεξιά
- Ο pacman τρώει τα φρούτα orange(o) και apple (a)
- Ο pacman σπάει το block
- και πηγαίνει ένα βήμα πίσω μόνο όταν τρώει το orange

Οι τελεστές που αντιπροσωπεύουν αυτές της ενέργειες είναι

```
def can_move_left(state):           #αν ο pacman μπορεί να πάει αριστερά
    return not state[0][0]=='p'
```

```
def can_move_right(state):           #αν ο pacman μπορεί να πάει δεξιά
    return not state[len(state)-1][0]=='p'
```

```
def cant_move(state): # αν ο ρομπωτ δεν μπορεί να κουνηθεί
```

```

    for i in range(len(state))
        if state[i][0]=='p' and state[i][1]=='#':
            return 1

def can_eat(state):    #αν ο pacman μπορεί να φάει τα ακόλουθα φρούτα
    for i in range(len(state)):
        if state[i][0]=='p' and state[i][1]=='a':
            return 1

        if state[i][0]=='p' and state[i][1]=='o':
            return 1

def move_right(state):
    if cant_move(state):    #αν η συνθήκη αυτή υσχεί
        for i in range(len(state)):    #ο pacman σπαι το εμπόδιο
            if state[i][0]=='p' and state[i][1]=='#':    #και κινείται κανονικά δεξιά
                state[i][1]=''
            if can_move_right(state):
                for i in range(len(state)):
                    if state[i][0]=='p':
                        state[i][0]=''
                        state[i+1][0]='p'
                        return state
            else:
                return state

    else:    #αλλιώς κινείται κανονικά δεξιά
        if can_move_right(state):
            for i in range(len(state)):
                if state[i][0]=='p':
                    state[i][0]=''

```

```

        state[i+1][0]='p'
        return state
    else:
        return state

def eat(state):                                     # έλεγχος αν μπορεί να φάει τα φρούτα
    if can_eat(state):
        for i in range(len(state)):
            if state[i][0]=='p' and state[i][1]=='a':    #αν είναι απλο φρουτο το τρώει
                state[i][1]=''
                return state
        for i in range(len(state)):
            if state[i][0]=='p' and state[i][1]=='o':    #αν δεν είναι απλό πάει ένα βήμα
πίσω
                state[i][1]=''
                for i in range(len(state)):
                    if state[i][0]=='p':
                        state[i][0]=''
                        state[i-1][0]='p'
                        return state
                return state
    else:
        return state

def find_children(state):
    children=[]
    right_state=copy.deepcopy(state)    #αντιγραφή τρέχουσας κατάστασης
    child_right=move_right(right_state)    #και χρήση τελεστών μετακίνησης
    left_state=copy.deepcopy(state)    #δεξιάς και χρήση φαγώματος.
    eat_state=copy.deepcopy(state)    #και δημιουργία της νέας κατάστασης
    child_eat=eat(eat_state)

```

```
if not child_eat==state:           #αν ο pacman μπορεί να κινηθεί
    children.append(child_eat)      #ή να φάει το φρούτο
                                   #προσθέτει την νέα κατάσταση στην λίστα
if not child_right==state:         #και επειστρέφει τα παιδιά
    children.append(child_right)
return children
```

Για την επίλυση του προβλήματος χρησιμοποιήθηκε η μεθοδος DFS και ο αλγόριθμος `search_algorithm_front_queue` που δώθηκε με το παραδειγμα της κυρίας Κολλία και τροποποιήθηκε καταληλα για να περιγραψει το δικο μας πρόβλημα .