Dfs and Bfs Explanation

BnB with Depth First Search:

class ItemValue:

Ορισμός των αντικειμένων weight, value, index, cost (value/weight) και ορισμός μεθόδου __lt__ για να μπορεί να γίνει σύγκριση μεταξύ αντικειμένων.

def getMaxValue:

Δέχεται σαν ορίσματα 2 λίστες wt(weights), val(values) και το capacity και υπολογίζει το νέο estimation κάθε φορά που δημιουργείται ένα δεξί παιδί.

Επίσης στη μεταβλητή bv κρατιέται κάθε φορά ένα νέο best value από τα αντικείμενα που χωράνε ολόκληρα μέσα στον σάκο και αλλάζει το current_solution αν η bv είναι μεγαλύτερη.

Επιστρέφεται η totalValue που είναι το νέο estimation

def newEstimation:

Ο σκοπός της συνάρτησης αυτής είναι να περαστεί η σωστή λίστα στη getMaxValue για τον υπολογισμό του νέου estimation. Η λίστες new_values και new_weights αποθηκεύουν τις αξίες και τα βάρη από τα αντικείμενα που έχουν μπει στον σάκο μέχρι και το συγκεκριμένο επίπεδο (αντικείμενο) που βρίσκεται στο δεξί κόμβο από τον οποίο καλείται η συνάρτηση καθώς και τα αντικείμενα μετά από αυτό.

Αν είμαστε στον πρώτο κόμβο τότε:

New_list = όλα τα αντικείμενα εκτός από το πρώτο

αλλιώς:

Για κάθε δείκτη στην check_list:

Αν ο δείκτης είναι μικρότερος από το τρέχον αντικείμενο:

New_list.append(το αντικείμενο με αυτό τον δείκτη)

Αν δεν είμαστε στον τελευταίο κόμβο:

new_list.append(όλα τα items μετά από το current_item)

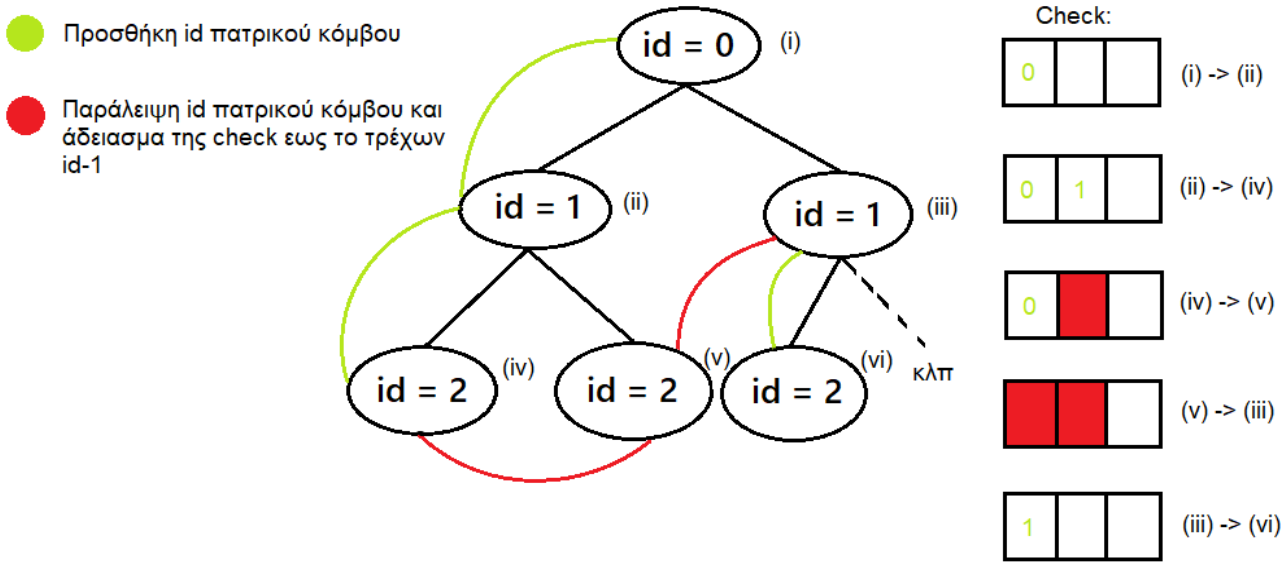
Main:

Η επίλυση του dfs έγινε με χρήση στοίβας όπου σε κάθε επανάληψη εξάγεται το τελευταίο στοιχείο προκειμένου να προσομοιωθεί η διάσχιση preorder.

(check = λίστα που περιλαμβάνει τους δείκτες των αντικειμένων που έχουν τοποθετηθεί έως το current_node (δηλαδή id αριστερών κόμβων))

Τα id των κόμβων αντιστοιχούν στους δείκτες της λίστας των αντικειμένων

Αν το current_node(id) < prev_id:
τότε μεταφερόμαστε σε χαμηλότερο επίπεδο άρα χρειάζεται άδειασμα η check από το current_id-1 και μετά.



Έλεγχος για κλάδεμα λόγω βελτιστοποίησης

Αν current_solution <= upper_bound(estimation)

Έλεγχος για πιθανή λύση

Αν total_value < upper_bound:

Δημιουργία αριστερού και δεξιού παιδιού

Stack.append(right_node)

Αν χωράει το αριστερό:

Stack.append(left_node)

Αλλιώς:

Αποθήκευση πιθανής λύσης

previous_id = current_id για την σύγκριση στην επόμενη επανάληψη

BnB with Best First Search:

Main:

Η επίλυση του Bfs έγινε με χρήση ουράς και γίνεται εξαγωγή του πρώτου στοιχείου μετά από ταξινόμηση των κόμβων με βάση το upper_bound (estimation)

Επειδή τώρα δεν γνωρίζουμε με ακρίβεια τον τρόπο διάσχισης του δέντρου δεν μπορούμε να φτιάξουμε την λίστα check με βάση τον προηγούμενο τρόπο.

Αντ' αυτού φτιάχνουμε μία νέα λίστα check για κάθε κόμβο (current_check).