

Η συνάρτηση βασίζεται σε 2 αρχές.

1.) Όποτε το φανάρι πάει απέναντι αυτός που θα το επιστρέψει θα πρέπει να είναι ένα από τα 2 πιο γρήγορα μέλη της οικογένειας. Αυτούς θα τους λέμε οδηγούς. Δεν χρειάζεται 3<sup>ος</sup> οδηγός γιατί ο χρόνος που παίρνει να πάνε 2 οδηγοί και να μείνουν απέναντι είναι ο ίδιος με τον χρόνο να μείνει ένας 2 φορές. Χρειάζεται πάνω από ένας γιατί γνωρίζουμε πως υπάρχουν περιπτώσεις όπου πρέπει να περάσουν 2 αργά μέλη μαζί για να ακυρωθεί ο χρόνος του σχετικά πιο γρήγορου όπως στο παράδειγμά μας. Σε αυτή τη περίπτωση πρέπει κάποιος να επιστρέψει το φανάρι και δεν έχει νόημα να είναι ένα από τα 2 αργά μέλη.

2.) Σε κάθε περίπτωση τα άτομα που θα περνάνε απέναντι θα είναι κατά προτεραιότητα τα πιο αργά ανά ζευγάρια ένας ένας μέχρι να ικανοποιηθεί η συνθήκη πως ο χρόνος A είναι πιο αργός από τον χρόνο B.

Χρόνος A = ο χρόνος που χρειάζεται για να περάσουν απέναντι 2 οδηγοί, να επιστρέψει ο ένας, να περάσουν μαζί τα 2 πιο αργά μέλη, και να επιστρέψει το φανάρι ο οδηγός που είχε μείνει εκεί, το ποιος μένει είναι αδιάφορο γιατί στο τέλος της μεταφοράς θα έχουν περάσει και οι 2 οδηγοί από 2 φορές ότι και να κάνουν, μια φορά μαζί μια μόνοι, οπότε θα μετρηθεί ο χρόνος του οδηγού B 2 φορές και του A μία.

Χρόνος B = χρόνος που χρειάζεται για να περάσουν τα επόμενα 2 πιο αργά μέλη ένας ένας συνοδευόμενα από τον γρήγορο οδηγό.

Το ποιος θα μείνει τελευταίο αργό μέλος δεξιά είναι αδιάφορο γιατί πάντα συμφέρει να τον μεταφέρει ο οδηγός 1.

Η συνθήκη εμφανίζεται έτσι:

```
2*(guide1+guide2)+family.get(i)<((2*guide1)+family.get(N-1-i)+family.get(N-2-i))
```

guide1-2 είναι οι χρόνοι των οδηγών και family.get(i) δίνει το πιο αργό μέλος που έχει μείνει δεξιά.

Η μόνη εκ πρώτης όψεως πιθανότητα να μην δίνει αυτή η στρατηγική βέλτιστη λύση είναι να πρέπει να περάσει ένα πολύ αργό μέλος μαζί με τον οδηγό 1 (μεταφορά τύπου 2) επειδή τα επόμενα ζευγάρια θα έχουν τέτοια μορφή (10.000 , 9.000), (9.000, 8.000), δηλαδή αν τα ζεύγη μεταφέρονταν κατά μία θέση το κάθε μέλος στον πάτο της λίστας θα ακύρωνε μεγαλύτερο χρόνο από ότι ακυρώνει τώρα, τον χρόνο του ακριβώς πιο πάνω στην ταξινομημένη λίστα χρόνου περάσματος. Αλλά αυτό δεν ωφελεί γιατί σε κάθε περίπτωση το κόστος του πιο αργού θα είναι μεγαλύτερο από το άθροισμα των διαφορών όλων των επόμενων ζευγαριών και το έξτρα κόστος οδήγησης των ζευγαριών είναι μεγαλύτερο οπότε δεν υπάρχει ούτε με αυτό σχετική οριακή κατάσταση και επιβεβαιώνεται πως πρέπει να περάσουν όλα τα αργά μέλη σαν ζευγάρια.

Η υλοποίησή της στο πρόγραμμα γίνεται ως εξής:

Δέχεται σαν είσοδο τον αριθμό των μελών της οικογένειας και τους χρόνους τους.

Αποθηκεύονται σε μια δυναμική λίστα που ταξινομείται.

Μετά εφαρμόζονται οι συνθήκες που επιλέγουν σε κάθε περίπτωση τη βέλτιστη μεταφορά.

Κάθε βήμα καθώς και ο τρόπος μεταφοράς εκτυπώνονται.

Πολλές εκτυπώσεις θα μπορούσαν να γενικοποιηθούν αλλά τις άφησα έτσι για να είναι πιο ευνόητος ο κώδικας.

Σημείωση: Αρχικά θα εφαρμοζόταν αλγόριθμος αναζήτησης με χρήση αυτής της συνάρτησης σαν ευρετικής η οποία θα περιόριζε το μέτωπο αναζήτησης σε ταξινομημένες μεταφορές ζευγαριών η ατόμων με διαφορετικούς συνδυασμούς και κάθε Node θα επέστρεφε το σύνολο των μεταφορών σε μορφή String τύπου (1 3 R, 3 L, 12 8 L, 1 R.....), αλλά όταν συνειδητοποίησα πως δεν υπήρχε περίπτωση αποτυχίας της (πέρα από σφάλμα στις συνθήκες ορίων επαναλήψεων) και μιας και η πρώτη άσκηση σε αντίθεση με τη δεύτερη δεν ζητάει συγκεκριμένο αλγόριθμο αλλά εξετάζει τον χρόνο εκτέλεσης απλοποίησα το πρόγραμμα ώστε να υλοποιείται σε  $O(N^2)$ .

Έξοδος εκτέλεσης με το παράδειγμα του βίντεο.

How many members does the family have?

5

Enter the time of a family member.

1

Enter the time of a family member.

3

Enter the time of a family member.

6

Enter the time of a family member.

8

Enter the time of a family member.

12

transfer type: 1

Family members of times 1 and 3 crossed to the left

time passed: 3

Family member of time 1 crossed back to the right

time passed: 4

Family members of times 12 and 8 crossed to the left

time passed: 16

Family member of time 3 crossed back to the right

time passed: 19

Transfer type: 2

Family members of times 6 and 1 crossed to the left

time passed: 25

Family member of time 1 crossed back to the right

time passed: 26

Family members of times 1 and 3 crossed to the left

29

Το πρόγραμμα δεν εμφανίζει χρόνους εκτέλεσης στην τάξη των δευτερολέπτων για πρακτικές εισόδους. Μέχρι 10.000 εισόδους εκτελείται σε λιγότερο από 1 δεύτερο.

Για παράδειγμα.

.....

Family members of times 999 and 999 crossed to the left

time passed: 19975005

Family member of time 999 crossed back to the right

time passed: 19976004

Family members of times 999 and 999 crossed to the left

19977003