

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ
6^ο εξάμηνο

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ
21/3/2020

Freecell Solitaire

Εισαγωγικά:

Έστω το παιχνίδι με κάρτες τύπου "πασιέντζα" με όνομα Freecell Solitaire, που διανέμεται μαζί με τα MS-Windows.



Μπορείτε να βρείτε μια ηλεκτρονική έκδοση του παιχνιδιού, μαζί με τους κανόνες του, στη διεύθυνση:

<https://cardgames.io/freecell/>

Συνοπτικά ισχύουν τα εξής:

- Έχουμε S στοίβες στο κάτω μέρος του παιχνιδιού. Αρχικά οι 52 κάρτες κατανέμονται τυχαία σε αυτές τις στοίβες. Για τη συνήθη τιμή $S=8$ και για 52 κάρτες, οι 4 πρώτες στοίβες να έχουν από 7 κάρτες και οι 4 τελευταίες από 6 κάρτες, όπως φαίνεται και στην παραπάνω εικόνα.

- Υπάρχουν τέσσερις ελεύθερες θέσεις πάνω αριστερά, κάθε μία από τις οποίες μπορεί να δεχθεί μία μόνο κάρτα.
- Υπάρχουν τέσσερις θέσεις πάνω δεξιά (foundations) όπου πρέπει να μεταφερθούν όλες οι κάρτες.

Οι επιτρεπτές κινήσεις ακολουθούν τους εξής κανόνες:

- Μπορούμε να μετακινούμε μία κάρτα τη φορά.
- Μια κάρτα μετακινείται όταν δεν έχει άλλη κάρτα πάνω της (είναι ελεύθερη). Έτσι, οι κάρτες των freecells μπορούν να μετακινηθούν, όπως και οι κορυφαίες κάρτες στις στοίβες.
- Η κάρτα που μετακινείται μπορεί να πάει σε μία από τις ακόλουθες θέσεις:
 - ο Σε ένα ελεύθερο freecell.
 - ο Στην κορυφή μιας στοίβας, εφόσον η προηγούμενη κορυφαία κάρτα της στοίβας είχε διαφορετικό χρώμα και κατά ένα μεγαλύτερη "αξία" (αριθμό).
 - ο Στη βάση μιας στοίβας, εφόσον η στοίβα είναι άδεια, χωρίς άλλο περιορισμό.
 - ο Στην κορυφή ενός foundation, εφόσον η προηγούμενη κορυφαία κάρτα του foundation είναι ίδιας "φυλής" και κατά ένα μικρότερης "αξίας" (αριθμό).
 - ο Στη βάση ενός άδειου foundation, εφόσον η κάρτα είναι "άσσος".

Μπορούμε να σκεφτούμε παραλλαγές του παραπάνω παιχνιδιού για λιγότερες κάρτες. Θα ορίσουμε ως N την αξία της μεγαλύτερης κάρτας, και θα θεωρήσουμε ότι έχουμε όλες τις κάρτες μέχρι αξία N (για το πλήρες παιχνίδι ισχύει $N=13$). Το πλήθος των φυλών, των freecells, των στοίβων και των foundations θα είναι πάντα το ίδιο, ανεξαρτήτως του πλήθους των καρτών.

Ζητούμενο αυτής της άσκησης είναι η κατασκευή ενός προγράμματος σε γλώσσα προγραμματισμού C ή C++ που θα λύνει προβλήματα freecell μεγέθους N , για τα διάφορες τιμές του N , χρησιμοποιώντας αλγορίθμους αναζήτησης.

Θεωρείστε το N ως σταθερά του προγράμματος (π.χ. `#define N 8`)

Θέμα 1^ο : Αλγόριθμοι απληροφόρητης (ή τυφλής) αναζήτησης

Κατασκευάστε ένα πρόγραμμα που να λύνει προβλήματα freecell με τους αλγορίθμους πρώτα σε πλάτος (breadth-first search) και πρώτα σε βάθος (depth-first search). Το πρόγραμμα θα δέχεται ως παραμέτρους τη μέθοδο επίλυσης, το όνομα του αρχείου περιγραφής του προβλήματος και το όνομα του αρχείου στο οποίο θα γραφεί η λύση. Για παράδειγμα, εάν το όνομα του προγράμματός σας είναι `freecell.exe` (μπορείτε φυσικά να το ονομάσετε όπως αλλιώς θέλετε), θέλετε να χρησιμοποιήσετε αναζήτηση κατά πλάτος, το αρχείο εισόδου είναι το `input.txt` και θέλετε η λύση να γραφεί στο αρχείο `solution.txt`, θα πρέπει να καλέσετε το πρόγραμμά σας με την εντολή:

```
freecell.exe breadth input.txt solution.txt
```

Εάν αντίθετα θέλετε να χρησιμοποιήσετε τον αλγόριθμο πρώτα σε βάθος, χρησιμοποιείτε τη λέξη `depth` αντί της λέξης `breadth` στην παραπάνω κλήση.

Σε σχέση με το αρχείο εισόδου `input.txt`, έστω ότι $N=13$, οπότε έχουμε 52 κάρτες και ότι θέλουμε να λύσουμε το πρόβλημα της αρχικής εικόνας. Συμβολίζουμε τις 13 αξίες με τους αριθμούς από το 1 έως το 13 και τις τέσσερις φυλές με τα γράμματα S (spades, ♠), H (hearts, ♥), D (diamonds, ♦) και C (clubs, ♣). Έτσι, για $N=13$ τα περιεχόμενα του αρχείου εισόδου για το πρόβλημα που φαίνεται στην αρχική εικόνα θα ήταν τα εξής:

```
H1 S2 C6 S11 C11 H4 C2
```

```
C4 D11 S4 C9 C12 H13 D4
```

```
H10 S6 D1 H12 S3 S12 D12
```

```
D5 S1 D8 C10 S7 S9 H6
```

```
H2 C5 H7 S10 D2 H11
```

C7 H5 D9 C8 D13 C13
H9 D6 C1 S5 D7 S13
D3 H3 C3 H8 D10 C8

Δηλαδή, το αρχείο έχει οκτώ (8) γραμμές, μια για κάθε αρχική στοίβα, ενώ σε κάθε γραμμή αναφέρονται οι κάρτες που βρίσκονται αρχικά σε αυτή τη στοίβα, πρώτα η φυλή και μετά (χωρίς κενό) η αξία. Ανάμεσα σε κάθε δύο κάρτες έχει ένα κενό (space). Σε περίπτωση που έχουμε λιγότερες κάρτες (π.χ. N=8), αυτές μοιράζονται και πάλι μεταξύ των στοιβών.

Σε σχέση με το αρχείο εξόδου solution.txt, το περιεχόμενό του θα πρέπει να είναι της παρακάτω μορφής (ακολουθεί ως παράδειγμα η λύση του τρέχοντος προβλήματος):

K
freecell C2
freecell D12
freecell S12
stack S3 H4
stack H12 C13
source D1
stack D12 S13
κλπ

όπου στην πρώτη γραμμή αναγράφεται το πλήθος των κινήσεων (π.χ. K=123), ενώ ακολουθούν οι κινήσεις, μια σε κάθε γραμμή. Οι δυνατές κινήσεις είναι οι εξής:

- freecell κάρτα
δηλώνοντας ότι η κάρτα μεταφέρεται σε ένα άδειο freecell.
- stack κάρτα1 κάρτα2
δηλώνοντας ότι η κάρτα1 μεταφέρεται πάνω στην κάρτα2 σε κάποια στοίβα.
- newstack κάρτα
δηλώνοντας ότι η κάρτα μεταφέρεται σε μία κενή θέση στοίβας, δημιουργώντας νέα στοίβα
- foundation κάρτα
δηλώνοντας ότι η κάρτα ανεβαίνει σε κάποιο foundation.

Το πρόγραμμά σας μπορεί να τυπώνει περιορισμένης έκτασης μηνύματα στην οθόνη, όπως π.χ. το χρόνο που χρειάστηκε για να λύσει το πρόβλημα, το πλήθος των βημάτων που έχει η λύση, ενδεχόμενα μηνύματα λάθους (π.χ. αδυναμία επίλυσης του προβλήματος μέσα σε συγκεκριμένα χρονικά όρια, π.χ. 5 mins κλπ).

Δώστε ιδιαίτερη προσοχή στον τρόπο κλήσης του προγράμματός σας, καθώς και στη μορφή των αρχείων εισόδου και εξόδου, σύμφωνα με όσα περιγράφηκαν παραπάνω, ώστε το πρόγραμμά (συμπεριλαμβανομένων των λύσεων που αυτό παράγει) να μπορεί να **ελεγχθεί αυτόματα**.

Προσοχή: Για την αποφυγή ατέρμονων βρόχων (αφορά κατά κύριο λόγο την αναζήτηση πρώτα σε βάθος), το πρόγραμμά σας θα πρέπει να ελέγχει για κύκλους τουλάχιστον στο τρέχον μονοπάτι.

Θέμα 2° : Αλγόριθμοι πληροφορημένης (ή ευρετικής) αναζήτησης

Η μοναδική διαφορά (σε επίπεδο υλοποίησης) των αλγορίθμων αναζήτησης πρώτα σε πλάτος και πρώτα σε βάθος είναι ότι ο πρώτος τοποθετεί τους νέους (γκρι) κόμβους στο τέλος του μετώπου (frontier) αναζήτησης ενώ ο δεύτερος στην αρχή. Υπάρχουν ωστόσο και άλλες επιλογές, οι οποίες οδηγούν σε ολόκληρη κατηγορία νέων αλγορίθμων. Η βασική εναλλακτική επιλογή είναι οι νέοι κόμβοι να μην τοποθετούνται ούτε στην αρχή, ούτε στο τέλος του μετώπου αναζήτησης, αλλά ενδιάμεσα. Ειδικότερα, κάθε νέος (γκρι) κόμβος βαθμολογείται με μια ευρετική συνάρτηση, η οποία παρέχει μια εκτίμηση της απόστασης (πλήθος κινήσεων) του κόμβου από την επιθυμητή τελική κατάσταση, ενώ το μέτωπο αναζήτησης διατηρείται ταξινομημένο με τους κόμβους που έχουν μικρότερη ευρετική τιμή πρώτους.

Σχεδιάστε μια ευρετική συνάρτηση για το συγκεκριμένο πρόβλημα. Η συνάρτησή σας θα πρέπει να λαμβάνει υπόψη διάφορους παράγοντες, όπως πόσες κάρτες έχουν ανεβεί στα foundations, πόσα freecells είναι ελεύθερα, πόσες στοίβες είναι κενές, κλπ. Προσπαθείστε η ευρετική σας συνάρτηση να μην υπερεκτιμά το πραγματικό πλήθος των κινήσεων, αν και κάτι τέτοιο δεν είναι απαραίτητο.

Υλοποιείστε λοιπόν τους παρακάτω δύο αλγορίθμους. Χρησιμοποιείστε τις λέξεις `best` και `astar` στην είσοδο για να τους δηλώσετε.

Αναζήτηση πρώτα στο καλύτερο (best-first search)

Βαθμολογείστε τους κόμβους με βάση την ευρετική σας συνάρτηση. Εισάγετε τους νέους κόμβους στην κατάλληλη θέση του μετώπου αναζήτησης, ώστε αυτό να παραμένει ταξινομημένο. Εξάγετε από το μέτωπο αναζήτησης προς εξέταση πάντα τον πρώτο κόμβο.

Αναζήτηση A* (A* search)

Παρόμοια με την αναζήτηση πρώτα στο καλύτερο, η μοναδική διαφορά της αναζήτησης A* είναι ότι στον βαθμό κάθε κόμβου συνυπολογίζεται (προστίθεται) και το πόσες κινήσεις εκτελέσατε από την αρχική κατάσταση για να φτάσετε στον τρέχοντα κόμβο. Για παράδειγμα, εάν ένας κόμβος έχει προκύψει ύστερα από 4 μετακινήσεις καρτών (σε σχέση με την αρχική διάταξη), ενώ η ευρετική απόσταση του από την τελική διάταξη είναι 90, ο συνολικός βαθμός του κόμβου είναι 94. Στην αναζήτηση A* το μέτωπο αναζήτησης διατηρείται ταξινομημένο με βάση το παραπάνω άθροισμα.

Θέμα 3^ο : Πειραματική αξιολόγηση των αλγορίθμων

Δοκιμάστε τους τέσσερις αλγορίθμους αναζήτησης που υλοποιήσατε σε διάφορα προβλήματα (συμπεριλαμβανομένων προβλημάτων διαφορετικών μεγεθών). Συγκρίνετε τους χρόνους επίλυσης και το πλήθος των βημάτων σε κάθε λύση. Εκθέστε τα συμπεράσματά σας όσον αφορά (συγκριτικά) το χρόνο που απαιτείται από κάθε αλγόριθμο για να λύσει τα κάθε πρόβλημα και το μήκος της λύσης που αυτός βρίσκει.

Δίνονται:

Σας δίνεται το πρόγραμμα `generator.c` το οποίο παράγει τυχαία αρχεία προβλημάτων συγκεκριμένης διάστασης (ορίζεται από την σταθερά `N` μέσα στο πρόγραμμα). Εάν το εκτελέσιμο ονομάζεται `generator.exe`, τότε το καλείτε με:

```
generator.exe <filename> <id1> <id2>
```

όπου `<filename>` το πρόθεμα των ονομάτων των αρχείων που θα δημιουργηθούν, και `<id1>` και `<id2>` δύο αριθμοί που δηλώνουν το εύρος των καταλήξεων των ονομάτων των αρχείων. Για παράδειγμα, εάν καλέσουμε το πρόγραμμα με:

```
generator.exe test 21 30
```

θα δημιουργηθούν 10 αρχεία προβλημάτων με ονόματα από `test21.txt` έως `test30.txt`.

Κριτήρια αξιολόγησης:

Θέμα 1° : Τυφλοί αλγόριθμοι αναζήτησης	
Αναζήτηση πρώτα σε πλάτος	20
Αναζήτηση πρώτα σε βάθος	20
Θέμα 2° : Ευρετικοί αλγόριθμοι αναζήτησης	
Αναζήτηση πρώτα στο καλύτερο	20
Αναζήτηση A*	10
Θέμα 3° : Υπολογιστική μελέτη	20
Γενική εικόνα (ευανάγνωστος κώδικας, σχολιασμός, κλπ.)	10
ΣΥΝΟΛΟ	100

Ο συνολικός βαθμός θα διαιρεθεί δια 100, ώστε να προκύψει ο τελικός βαθμός της εργασίας (με μέγιστη τιμή το 1).

Οδηγίες υποβολής: Η εργασία θα πρέπει να υποβληθεί μέσω του classroom.google.com μέχρι την υποδεικνυόμενη ημερομηνία. Θα υποβάλλετε ένα έγγραφο κειμένου με περιγραφή των πειραμάτων σας και των αποτελεσμάτων σας, καθώς και τα αρχεία κώδικα και προβλημάτων που υλοποιήσατε/χρησιμοποιήσατε. Ο κώδικας θα πρέπει να είναι ευανάγνωστος (προσοχή στη χρήση των εσοχών) και επαρκώς σχολιασμένος. Μπορείτε να χρησιμοποιήσετε οποιαδήποτε γλώσσα προγραμματισμού θέλετε.