

# Δομές Δεδομένων και Αλγόριθμοι

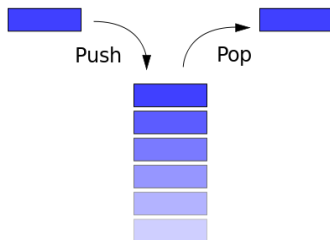
Χρήστος Γκόγκος

ΤΕΙ Ηπείρου

Χειμερινό Εξάμηνο 2014-2015  
Παρουσίαση 14. Στοιίβες

Η στοίβα είναι μια ειδική περίπτωση γραμμικής λίστας στην οποία οι εισαγωγές και οι διαγραφές επιτρέπονται μόνο από το ένα άκρο. Συνήθως αυτό το άκρο λέγεται κορυφή (top)

Η στοίβα είναι μια δομή τύπου LIFO (Last In First Out)



- **empty()** ελέγχει αν οι στοίβα είναι άδεια.
- **size()** επιστρέφει το πλήθος των στοιχείων της στοίβας
- **push()** ωθεί ένα στοιχείο στην κορυφή της στοίβας.
- **pop()** αφαιρεί το στοιχείο που είναι στην κορυφή της στοίβας.
- **top()** επιστρέφει το στοιχείο που είναι στην κορυφή της στοίβα (χωρίς να το αφαιρεί).

# Στοιβες στη C++ std::stack

## Παράδειγμα

Ο ακόλουθος κώδικας εξετάζει αν ένα κείμενο είναι παλινδρομικό δηλαδή διαβάζεται το ίδιο από αριστερά προς τα δεξιά και από δεξιά προς τα αριστερά.

```
#include <stack>

...
stack<char> st;
char s[] = "amanaplanacanalpanama";
for (unsigned int i = 0; i < strlen(s); i++) {
    st.push(s[i]);
}
cout << "size= " << st.size() << endl;
char s2[st.size()];
int i = 0;
while (!st.empty()) {
    s2[i] = st.top();
    st.pop();
    i++;
}
s2[i]='\0';
if (strcmp(s, s2) == 0) cout << "Palindrome" << endl;
else cout << "Not a palindrome" << endl;
...
```

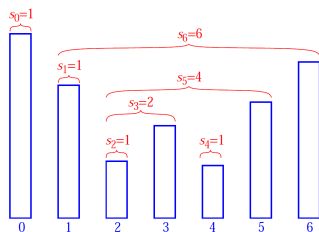
- Στοίβα χρόνου εκτέλεσης
- Υλοποίηση λειτουργιών Undo/Redo σε προγράμματα (π.χ. επεξεργαστής κειμένου)
- Λειτουργία ως βοηθητικής δομής για την υλοποίηση αλγορίθμων (π.χ. για την υλοποίηση Depth First Search σε γραφήματα)

# Στοίβα χρόνου εκτέλεσης (Runtime Stack)

Χρησιμοποιείται κατά την κλήση συναρτήσεων. Όταν μια συνάρτηση καλείται ωθείται στη στοίβα χρόνου εκτέλεσης ένα πλαίσιο δεδομένων που περιέχει στοιχεία που αφορούν τη συνάρτηση που κλήθηκε καθώς και τη διεύθυνση επιστροφής δηλαδή τη διεύθυνση στην οποία θα πρέπει να επιστρέψει η εκτέλεση όταν ολοκληρωθεί η κλήση της συνάρτησης. Όταν ολοκληρώνεται η κλήση της κάθε συνάρτησης το πλαίσιο το οποίο είναι στην κορυφή της στοίβας χρόνου εκτέλεσης απωθείται από τη στοίβα και ο έλεγχος περνά στη συνάρτηση που είναι πλέον στην κορυφή της λίστας.

# Εφαρμογή στοίβας: το πρόβλημα Stock Span

Στο Stock Span πρόβλημα δίνονται οι τιμές που σημείωσε μια μετοχή για έναν αριθμό ημερών και πρέπει να υπολογιστεί για κάθε ημέρα το span. Το span για μια ημέρα  $i$  ορίζεται ως ο μεγαλύτερος αριθμός από συνεχόμενες ημέρες μέχρι και την ημέρα  $i$  που η τιμή της μετοχής ήταν μικρότερη ή ίση από την τιμή της μετοχής στην ημέρα  $i$ . Για παράδειγμα για τις τιμές [100, 80, 60, 70, 60, 75, 85] οι span τιμές είναι [1, 1, 1, 2, 1, 4, 6]



# Λύση του προβλήματος stock span με τετραγωνική πολυπλοκότητα $O(n^2)$

```
void stock_span(int* stock_prices, int* spans, int N) {  
    for (int i = 0; i < N; i++) {  
        int c = 1;  
        for (int j = i - 1; j >= 0; j--) {  
            if (stock_prices[j] <= stock_prices[i]) {  
                c++;  
            } else  
                break;  
        }  
        spans[i] = c;  
    }  
}  
  
...  
int N = 7;  
int stock_prices[] = { 100, 80, 60, 70, 60, 75, 85 };  
int* spans = new int[N](); // initializes with zero  
stock_span(stock_prices, spans, N);  
for (int i = 0; i < 7; i++) printf ("%d ", spans[i]);  
delete[] spans;
```



# Λύση του προβλήματος stock span με γραμμική πολυπλοκότητα $O(n)$

```
void stock_span_using_stack(int* stock_prices, int* spans, int N) {
    stack<int> st;
    spans[0] = 1;
    st.push(0);
    for (int i = 1; i < N; i++) {
        while (!st.empty() && stock_prices[st.top()] <
               stock_prices[i])
            st.pop();
        if (st.empty())
            spans[i] = i + 1;
        else
            spans[i] = i - st.top();
        st.push(i);
    }
}

...
int N = 7;
int stock_prices[] = { 100, 80, 60, 70, 60, 75, 85 };
int* spans = new int[N](); // initializes with zero
stock_span_using_stack(stock_prices, spans, N);
for (int i = 0; i < 7; i++) printf ("%d ", spans[i]);
delete [] spans;
```

Γιατί η λύση του προβλήματος stock span με χρήση στοίβας έχει γραμμική πολυπλοκότητα  $O(n)$

Κάθε στοιχείο του πίνακα προστίθεται και αφαιρείται από την στοίβα το πολύ 1 φορά. Άρα οι λειτουργίες που γίνονται στην στοίβα είναι το πολύ  $2n$ . Καθώς κάθε λειτουργία στην στοίβα εκτελείται σε  $O(1)$  η πολυπλοκότητα χειρότερης περίπτωση είναι  $O(n)$

# Εφαρμογή στοίβας: infix και postfix μορφές αλγεβρικών παραστάσεων

Στην infix μορφή μιας παράστασης οι τελεστές (πράξεις) τοποθετούνται ανάμεσα από τους τελεστέους (σταθερές και μεταβλητές) (π.χ.  $x + y$ ).

Στην postfix μορφή μιας παράστασης οι τελεστές τοποθετούνται μετά από τους τελεστέους (π.χ.  $xy+$ ).

## Υπολογισμός postfix μορφής αλγεβρικής παράστασης με τη χρήση στοίβας

Η παράσταση επεξεργάζεται από αριστερά προς τα δεξιά και για κάθε σύμβολο:

Αν είναι μεταβλητή ή σταθερά ωθείται στη στοίβα

Αν είναι τελεστής τότε απωθούνται 2 στοιχεία από την κορυφή της στοίβας, εφαρμόζεται η πράξη που υποδηλώνει ο τελεστής και το αποτέλεσμα ωθείται στη στοίβα

π.χ. Η παράσταση  $3\ 4\ 1-2\ *\ +\ 7\ +$  ισούται με την τιμή 16 τα δε περιεχόμενα της στοίβας σε κάθε βήμα για να παραχθεί το αποτέλεσμα είναι  $[3]$ ,  $[3,4]$ ,  $[3,4,1]$ ,  $[3,3]$ ,  $[3,3,2]$ ,  $[3,6]$ ,  $[9]$ ,  $[9,7]$ ,  $[16]$ .

# Μετατροπή infix σε postfix

## Διαδικασία μετατροπής infix σε postfix

Η είσοδος είναι μια ακολουθία συμβόλων σε infix μορφή και η έξοδος είναι η ισοδύναμη ακολουθία σε postfix μορφή. Για κάθε σύμβολο της ακολουθίας εισόδου:

Αν είναι μεταβλητή ή σταθερά τοποθετείται στην παράσταση εξόδου απευθείας

Αν είναι τελεστής συγκρίνεται με το κορυφαίο στοιχείο της στοίβας και αν έχει υψηλότερη προτεραιότητα από αυτό εισάγεται στην κορυφή της στοίβας.

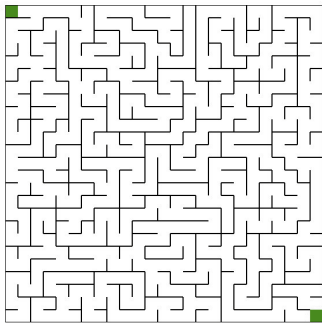
Αν έχει χαμηλότερη ή ίση προτεραιότητα τότε απωθούνται όλοι οι τελεστές με υψηλότερη προτεραιότητα τοποθετούνται στην παράσταση εξόδου και ο νέος τελεστής μπαίνει στη στοίβα.

π.χ. για την infix έκφραση  $a+b*c+d^e*z-w$  η postfix μορφή είναι  $abc^*+de^z^*+w-$

# Εφαρμογή στοίβας: Αναζήτηση μονοπατιού σε λαβύρινθο

## Λαβύρινθος

Είναι μια ορθογώνια περιοχή με μια είσοδο και μια έξοδο. Το εσωτερικό του περιέχει τοίχους που δεν μπορούν να διαπερασθούν. Το μονοπάτι είναι μια ακολουθία θέσεων από την είσοδο προς την έξοδο για τις οποίες ισχύει ότι καμία δεν είναι μπλοκαρισμένη.



- <http://www.geeksforgeeks.org/the-stock-span-problem/>