

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή ΗΜ&ΜΥ
Προγραμματιστικές Τεχνικές
2^ο εξάμηνο



Λύσεις Παλαιών Θεμάτων

Γερακάρης Βασίλης
<vgerak@gmail.com>
Α.Μ.: 03108092

1 Σεπτέμβριος 2007

1.1 Θέμα 1 °

Κάθε βασική λειτουργία (tfetch, tstore, tC, tH, κ.λ.π.) = 1 μονάδα χρόνου, να υπολογιστεί χρόνος εκτέλεσης και άνω ασυμπτωτικό όριο $O()$ για τον παρακάτω κώδικα:

```
1 int K = 0;
2 for (int i = 0; i < n; ++i)
3     for (int j = 1; j < n; ++j)
4         ++K;
```

1.2 Θέμα 2 °

Θεωρώ ότι η isEmpty επιστρέφει 1 αν η στοίβα είναι άδεια, αλλιώς επιστρέφει 0. Επιπλέον, η pop έχει ως τιμή επιστροφής την τιμή του στοιχείου που αφαιρείται.

```
1 int product (stack s)
2 {
3     int acc = 1;    //1 is the neutral element for multiplication
4     while (isEmpty(s))
5         acc *= pop(s);
6     return acc
7 }
```

```
1 //Easier to comprehend
2 int product (stack s)
3 {
4     int acc = 1;
5     int temp;
6     while (isEmpty(s) != 1) {
7         temp = pop(s);
8         acc = acc * temp
9     }
10    return acc
11 }
```

1.3 Θέμα 3 °

Cheat answer: Δε θα τυπωθεί τίποτα, θα φάει syntax error στην "printf("ΤΕΛΟΣ \n");

Serious answer:

A)

- f(10) -> τυπώνει 10, μπαίνει στο "n%2 == 0" (c)
- f(5) -> τυπώνει 5, μπαίνει στο "else" (c)
- f(16) -> τυπώνει 16, μπαίνει στο (b)
- f(8) -> τυπώνει 8, μπαίνει στο (b)
- f(4) -> τυπώνει 4, μπαίνει στο (b)
- f(2) -> τυπώνει 2, μπαίνει στο (b)
- f(1) -> τυπώνει 1, μπαίνει στο "n==1", τυπώνει "ΤΕΛΟΣ" και τερματίζει

B) Θα τυπωθούν ακριβώς τα ίδια μηνύματα, με αντίστροφη σειρά:

ΤΕΛΟΣ, 1, 2, 4, 8, 16, 5, 10

1.4 Θέμα 4 °

Υποθέτουμε ότι οι `stdlib.h` και `string.h` έχουν γίνει `include` στο αρχείο μας.
Υποθέτουμε επίσης ότι η δομή που χειριζόμαστε είναι η παρακάτω.

```
1 struct node {
2     char *word;
3     struct node *next;
4 };

1 struct node *remDupl(struct node *myList)
2 {
3     struct node *ptr;
4     struct node *temp;
5     if (myList == NULL)
6         return myList;  \\check if empty list
7     ptr = myList;
8     while (ptr->next != NULL) {
9         if (strcmp(ptr->word, ptr->next->word) == 0) {\{ \\if we find string match
10             temp = ptr->next->next; \\keep where the 2nd node points
11             free(ptr->next);          \\delete node
12             ptr->next = temp;          \\restore pointer
13         } else
14             ptr = ptr->next;          \\check next 2 nodes
15     }
16 }
```

1.5 Θέμα 5 °

A)

```
1 int c = 0;          //global
2
3 int traverseCount(tree t)
4 {
5     if(t == NULL)
6         return 0;
7     if(t->left != NULL && t->right != NULL)
8         ++c;
9     traverseCount(t->left);
10    traverseCount(t->right);
11    return c;
12 }
```

B) 5 -> 4 -> 2 -> 8 -> 6 -> 7 -> 9

1.6 Θέμα 6 °

Αρχικά : [2,1,4,1,6,8,3,5]

i = 7:

j = 1 : [1,2,4,1,6,8,3,5] (swap 1,2)

j = 2 : [1,2,4,1,6,8,3,5]

j = 3 : [1,2,1,4,6,8,3,5] (swap 1,4)

j = 4 : [1,2,1,4,6,8,3,5]

j = 5 : [1,2,1,4,6,8,3,5]

j = 6 : [1,2,1,4,6,3,8,5] (swap 3,8)

j = 7 : [1,2,1,4,6,3,5,8] (swap 5,8)

i = 6:

j = 1 : [1,2,1,4,6,3,5,8]

j = 2 : [1,1,2,4,6,3,5,8] (swap 1,2)

j = 3 : [1,1,2,4,6,3,5,8]

j = 4 : [1,1,2,4,6,3,5,8]

j = 5 : [1,1,2,4,3,6,5,8] (swap 3,6)

j = 6 : [1,1,2,4,3,5,6,8] (swap 5,6)

i = 5:

j = 1 : [1,1,2,4,3,5,6,8]

j = 2 : [1,1,2,4,3,5,**6**,8]

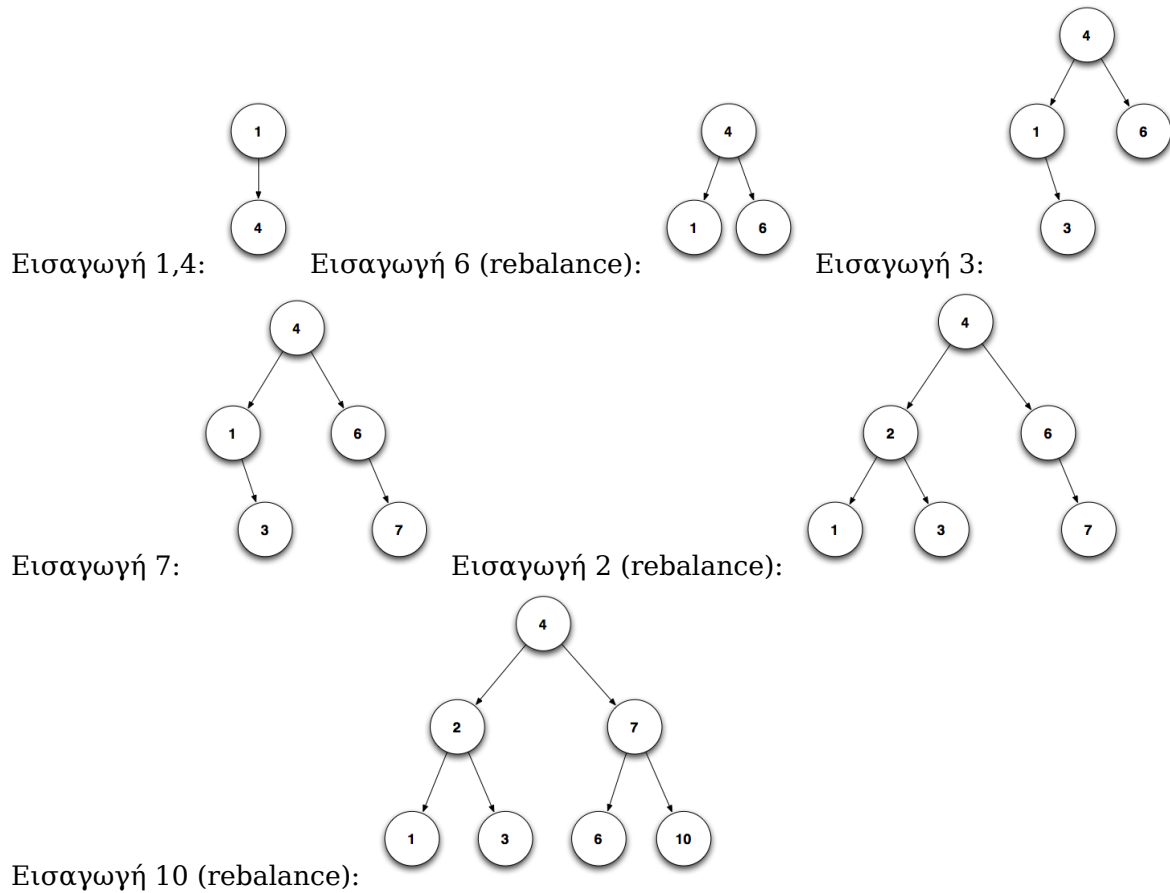
j = 3 : [1,1,2,4,3,5,**6**,8]

j = 4 : [1,1,2,3,4,5,**6**,8] (swap 3,4)

j = 5 : [1,1,2,3,4,**5**,6,8]

i = 4: Καμία αντιμετάθεση, ο πίνακας είναι ταξινομημένος.

1.7 Θέμα 7 °



2 Ιούλιος 2011