

# Συναρτησιακός Προγραμματισμός 2008

## Τελική Εξέταση

Η εξέταση διαρκεί 2 ώρες και αντιστοιχεί στο 55% του βαθμού του μαθήματος.

**Συνολικό Άθροισμα Βαθμών: 550.**

**Επιτρέπονται: Σημειώσεις σχετικά με το μάθημα.**

**Όλα τα προγράμματα να γραφούν σε Haskell.**

**Καλή Επιτυχία!**

1. **Βαθμοί: 10.** Γράψτε μία συνάρτηση με όνομα `encode` που να παίρνει ως όρισμα μία λίστα `l`, και την κωδικοποιεί ως εξής:

- κάθε μεγίστου-μήκους κομμάτι της `l` που περιέχει  $n$  εμφανίσεις του ίδιου αντικειμένου `x` κωδικοποιείται από ένα ζεύγος  $(x, n)$ .
- ολόκληρη η `l` κωδικοποιείται ως λίστα τέτοιων ζευγών.

Για παράδειγμα, η έκφραση

```
encode [2, 2, 3, 3, 3, 3, 2, 2, 2, 6, 3, 3]
```

θα πρέπει να αποτιμηθεί σε

```
[(2, 2), (3, 4), (2, 3), (6, 1), (3, 2)]
```

Ποιος είναι ο πιο γενικός πολυμορφικός τύπος που περιγράφει τη συνάρτησή σας;

2. **Βαθμοί: 15.** Το πρόβλημα των βασιλισσών περιγράφεται ως εξής: έστω μία σκακιέρα  $n \times n$  όπου  $n$  ακέραιος με  $n > 3$ . Ζητείται να τοποθετηθούν  $n$  βασίλισσες σκακιού στη σκακιέρα, έτσι ώστε καμία να μην απειλεί την άλλη. Μία βασίλισσα απειλεί μία άλλη αν οι δύο βασίλισσες βρίσκονται στην ίδια σειρά, στήλη ή διαγώνιο. Χρησιμοποιώντας όποια αναπαράσταση δεδομένων θέλετε, γράψτε μία συνάρτηση με όνομα `queens` η οποία να παίρνει έναν ακέραιο  $n$  και να επιστρέφει μία λίστα με *όλες* τις λύσεις του προβλήματος σε σκακιέρα  $n \times n$ .

**Υπόδειξη:** Χρησιμοποιήστε εκφράσεις διαχωρισμού.

3. **Βαθμοί: 15.** Θέλουμε να αναπαραστήσουμε τους *απλούς μη κατευθυνόμενους γράφους χωρίς βάρη*. Δηλαδή, ένας γράφος αντιπροσωπεύεται από ένα σύνολο από *κόμβους*  $V$  μαζί με ένα σύνολο *ακμές*  $E \subseteq V^2$ . Για ένα γράφο με  $n$  κόμβους, όπου  $n \in \mathbb{N}$ , θα θεωρούμε ότι οι κόμβοι του είναι όλοι οι φυσικοί από το 0 έως το  $n - 1$ .

Χρησιμοποιώντας όποια αναπαράσταση δεδομένων θέλετε, υλοποιήστε ένα `ATA Graph` που να αναπαριστά τέτοιους γράφους, μαζί με συναρτήσεις τέτοιες ώστε:

- Να είναι δυνατόν να κατασκευαστούν *όλοι* οι γράφοι (συμπεριλαμβανομένου και του μηδενικού γράφου) και *μόνον αυτοί*.
- Δωθέντος ενός γράφου, να μπορούμε να πάρουμε κάθε πληροφορία που θέλουμε από αυτόν, δηλ. ποιο είναι το μέγεθός του και αν μία ακμή είναι ή δεν είναι στο γράφο.

**Υπόδειξη:** Προσέξτε το γεγονός ότι ο γράφος είναι *μη κατευθυνόμενος*.

4. **Βαθμοί: 15.** Υλοποιήστε μία μονάδα `AppCount` η οποία να μετράει τον αριθμό εφαρμογών συναρτήσης στην αποτίμηση μίας έκφρασης. Εφαρμογές της `return` θα πρέπει να συμπεριλαμβάνονται σε αυτόν τον αριθμό.

Γράψτε μία συνάρτηση Haskell με όνομα `appcount` και τύπο `AppCount a -> Int`

που να επιστρέφει αυτόν τον αριθμό.

Για παράδειγμα, έστω οι παρακάτω ορισμοί

```
fact :: Int -> AppCount Int
fact 0 = do return 1
fact n = do x <- fact(n-1)
          return (x*n)
sc :: Int->AppCount Int
sc n = do return (n+1)
```

Η αποτίμηση της έκφρασης

```
appcount (do x<-fact 3 ; sc x ; return x)
```

πρέπει να επιστρέφει 6.