

Οι ασκήσεις αυτές είναι για να λυθούν από εσάς. Αν έχετε πρόβλημα να λύσετε κάποια άσκηση ζητείστε βοήθεια στο Forum του μαθήματος. Οι λύσεις θα δημοσιεύονται 1-2 βδομάδες μετά από την ανάρτηση του κάθε Φυλλαδίου ασκήσεων.

1. Για σταθερό  $n$  ποιο (ή ποια) είναι το  $k$  για το οποίο μεγιστοποιείται ο διωνυμικός συντελεστής  $\binom{n}{k}$ ;
2. Σε μια πόλη με εξαψήφια τηλέφωνα πόσα νούμερα το πολύ μπορεί να υπάρχουν χωρίς επαναλαμβανόμενα ψηφία;
3. Ένα μήνυμα στον κώδικα Morse είναι μια πεπερασμένη ακολουθία (μια λέξη όπως λέμε) από κουκίδες, παύλες και κενά. Πόσα διαφορετικά μηνύματα φτιάχνονται με 7 κουκίδες, 3 παύλες και 2 κενά;
4. Η νεοσύστατη εταιρεία πληροφορικής «Faster Than Light» ετοιμάζεται να φτιάξει ένα σύστημα για online email το οποίο φιλοδοξεί να καλύψει στο μέλλον πάνω από 100 εκατομμύρια χρήστες. Η εταιρεία έχει αποφασίσει ότι όλες οι διευθύνσεις email των χρηστών της θα είναι της μορφής

`username@does-not-get-there.ever`

όπου το username του χρήστη θα απαρτίζεται από το πολύ  $N$  γράμματα από το σύνολο

A, B, C, D, ..., Z, 0, 1, ..., 9

(36 σύμβολα συνολικά) με τον περιορισμό ότι το πρώτο γράμμα του `username` πρέπει να είναι γράμμα και όχι αριθμός. (Τα κεφαλαία γράμματα ταυτίζονται με τα μικρά.) Ποια είναι η ελάχιστη τιμή του  $N$  που πρέπει να προβλέψει η εταιρεία ώστε να μπορεί να ικανοποιήσει το στόχο της και να έχει τουλάχιστον 100 εκατομμύρια διαθέσιμες διευθύνσεις στην αρχή της λειτουργίας της;

5. Αποδείξτε ότι  $\frac{n!}{2^n} \rightarrow \infty$ , για  $n \rightarrow \infty$ .



Στο κλάσμα  $\frac{1 \cdot 2 \cdot 3 \cdot \dots \cdot n}{2 \cdot 2 \cdot 2 \cdot \dots \cdot 2}$  μπορείτε να διαγράψετε τους μισούς όρους από αριστερά πάνω και κάτω και το κλάσμα να μικράνει. Όλοι οι παράγοντες που απομένουν στον αριθμητή είναι  $\geq n/4$ .

6. Αποδείξτε, χωρίς να κάνετε τις πράξεις, και μόνο ερμηνεύοντας ως ένα αριθμό επιλογών το αριστερό και το δεξί μέλος την ταυτότητα

$$\binom{n}{k} = \binom{n}{n-k}, \quad \text{για } 0 \leq k \leq n.$$

7. Αποδείξτε, χωρίς να κάνετε τις πράξεις, την ταυτότητα

$$2^n = \sum_{k=0}^n \binom{n}{k}.$$



Το αριστερό μέλος,  $2^n$ , μετράει πόσα υποσύνολα, οποιοδήποτε μεγέθους, έχει το σύνολο  $\{1, 2, \dots, n\}$ . Στο δεξί μέλος ο διωνυμικός συντελεστής  $\binom{n}{k}$  μετράει κι αυτός κάποιου είδους υποσύνολα.

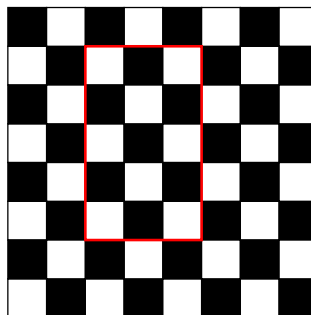
8. Από μια ομάδα 10 ατόμων, με πόσους τρόπους μπορεί να επιλεγεί ένα τριμελές προεδρείο χωρίς διακριτούς ρόλους; Ένα 5μελές προεδρείο με πρόεδρο, αντιπρόεδρο και 3 μέλη;



Για το δεύτερο ερώτημα, επιλέξτε το προεδρείο επιλέγοντας πρώτα τον πρόεδρο, μετά τον αντιπρόεδρο και, τέλος, τα τρία μέλη μαζί.

9. Μέ πόσους τρόπους μπορούμε να επιλέξουμε, από μια συνηθισμένη τράπουλα με 52 φύλλα (που χωρίζονται σε 4 χρώματα και 13 είδη), πέντε φύλλα από τα οποία 2 κόκκινα ( $\diamond$  ή  $\heartsuit$ ) και 3 σπαθιά; Δε μας ενδιαφέρει η σειρά επιλογής των φύλλων.

10. Σε μια  $8 \times 8$  σκακιέρα πόσα διαφορετικά ορθογώνια ορίζονται; Ένα ορθογώνιο είναι ένα υποσύνολο των κελιών (τετραγώνων) της σκακιέρας που έχει σχήμα ορθογωνίου. Δύο ορθογώνια θεωρούνται διαφορετικά αν είναι διαφορετικά ως σύνολα κελιών.



Μια  $8 \times 8$  σκακιέρα κι ένα από τα ορθογώνια τα οποία θέλουμε να μετρήσουμε

Δείτε στο Σχήμα.

11. (α) Με πόσους τρόπους μπορούμε να διατάξουμε τα ψηφία 1, 2, ..., 9 ώστε ανάμεσα στο 1 και το 2 να υπάρχουν ακριβώς τρία ψηφία;  
 (β) Με πόσους τρόπους μπορούμε να διατάξουμε τα ψηφία 1, 2, ..., 9 ώστε το 1 να προηγείται του 2 και το 2 να προηγείται του 3;

12. Η παρακάτω συνάρτηση Python υπολογίζει όλα τα υποσύνολα ενός πεπερασμένου συνόλου, το οποίο δίνεται ως μια λίστα. Η συνάρτηση είναι αναδρομική, καλεί δηλαδή τον εαυτό της. Πρώτα βρίσκει όλα τα υποσύνολα που δεν περιέχουν το πρώτο στοιχείο του συνόλου και μετά προσθέτει σε καθένα από αυτά το πρώτο στοιχείο του συνόλου. Έτσι προκύπτουν όλα τα υποσύνολα.

Αν δεν έχετε python 3 εγκατεστημένη τον υπολογιστή σας μπορείτε να τρέξετε αυτό το πρόγραμμα και να παίξετε με αυτό online, π.χ. στο <https://repl.it/@kolount/list-all-subsets>

```
def subsets(s):
    """Η συνάρτηση υπολογίζει και επιστρέφει όλα τα υποσύνολα του συνόλου
        s, το οποίο μας δίδεται ως μια λίστα
        , π.χ. ως τη λίστα ['a', 'b', 'c']. Το αποτέλεσμα μας επιστρέφεται ως μια λίστα από σύνολα λίστα
        ( από λίστες δηλ.).
    """
    n = len(s) # Το μέγεθος του συνόλου

    if n==0: # Το κενό σύνολο περιέχει μόνο το κενό [] ως υποσύνολο.
        return [[]]

    f = s[0] # Το f είναι το πρώτο στοιχείο του συνόλου s
    s1 = s[1:] # Το s1 είναι το σύνολο s χωρίς το πρώτο του στοιχείο

    p1 = subsets(s1) # Βρίσκουμε πρώτα όλα τα υποσύνολα του s που δεν περιέχουν το f.
                    # Αυτά είναι όλα τα υποσύνολα του s1

    p2 = [] # Στη λίστα p2 θα βάλουμε όλα τα υποσύνολα του s που περιέχουν το f.
    for x in p1:
        p2.append([f]+x) # Για κάθε υποσύνολο του s1 του προσθέτουμε το f

    return p2+p1 # Επιστρέφουμε την ένωση των δύο λιστών από σύνολα

print( subsets([1,2,3,4]) ) # Εδώ ζητάμε όλα τα υποσύνολα του συνόλου {1, 2, 3, 4}.
```

Το output του παραπάνω προγράμματος φαίνεται παρακάτω.

```
[
    [1, 2, 3, 4], [1, 2, 3], [1, 2, 4], [1, 2], [1, 3, 4], [1, 3], [1, 4], [1],
    [2, 3, 4], [2, 3], [2, 4], [2], [3, 4], [3], [4], []
]
```