Πιθανοτητες – Άνοιξη 2020-21

Λύσεις των ασκήσεων

1. Για σταθερό n ποιο (ή ποια) είναι το k για το οποίο μεγιστοποιείται ο διωνυμικός συντελεστής $\binom{n}{k}$;

Λύση: Έχουμε $\binom{n}{k} = \binom{n}{k-1}\frac{n-k+1}{k}$ και ο τελευταίος παράγοντας είναι ≥ 1 αν και μόνο αν $n+1 \geq 2k$. Άρα για n άρτιο η ακολουθία $\binom{n}{k}$ είναι αύξουσα μέχρι το $k = n/2$ (όπου έχουμε και το μέγιστο διωνυμικό συντελεστή) ενώ για n περιττό είναι αύξουσα μέχρι το $k = (n+1)/2$ και τη μέγιστη τιμή την παίρνει γι' αυτό το k και το αμέσως προηγούμενο.

2. Σε μια πόλη με εξαψήφια τηλέφωνα πόσα νούμερα το πολύ μπορεί να υπάρχουν χωρίς επαναλαμβανόμενα ψηφία;

Λύση: Αφού το τηλέφωνο είναι εξαψήφιο το πρώτο ψηφίο δε μπορεί να είναι 0, άρα έχουμε 9 επιλογές γι' αυτό. Για το 2ο ψηφίο (που δεν ισχύει ο περιορισμός πλέον του μη μηδενικού ψηφίου) έχουμε και πάλι 9 επιλογές (οποιοδήποτε από τα 10 ψηφία εκτός αυτό που επιλέξαμε για 1ο ψηφίο). Για το 3ο ψηφίο έχουμε 8 επιλογές, κλπ. Η τελική απάντηση είναι $9 \cdot 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5$.

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;

Λύση: Έχουμε $7+3+2 = 12$ θέσεις να γεμίσουμε με τα σύμβολα αυτά. Επιλέγουμε 7 από αυτές τις θέσεις για τις κουκίδες, και, από τις υπόλοιπες επιλέγουμε 3 για τις παύλες και έχουμε τελειώσει. Σύνολο $\binom{12}{7}\binom{5}{3}$.

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;

Λύση: Αν το username έχει k γράμματα τότε οι επιλογές μας είναι 26 για το πρώτο γράμμα και 36 για κάθε επόμενο, δηλ. $26 \cdot 36^{k-1}$. Αν το username έχει **το πολύ** N γράμματα τότε οι επιλογές μας είναι

$$26 \cdot 36^{N-1} + 26 \cdot 36^{N-2} + \cdots 26 \cdot 36 + 26 = 26(36^{N-1} + 36^{N-1} + \cdots + 36 + 1) = 26\frac{36^N - 1}{36-1} = 26\frac{36^N - 1}{35}.$$

Αν θέλουμε αυτός ο αριθμός να είναι $\geq 10^8$ τότε έχουμε ισοδύναμα $36 \geq 1 + 10^8\frac{35}{36} = 134615385.615384615$ και παίρνοντας λογαρίθμους βλέπουμε ότι πρέπει να ισχύει $N \geq 5.22$, οπότε πρέπει να πάρουμε $N = 6$.

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

Στο κλάσμα $\frac{1\cdot2\cdot3\cdots n}{2\cdot2\cdot2\cdots2}$ μπορείτε να διαγράψετε τους μισούς όρους από αριστερά πάνω και κάτω και το κλάσμα να μικράνει. Όλοι οι παράγοντες που απομένουν στον αριθμητή είναι $\geq n/4$.

Λύση: Με βάση την υπόδειξη έχουμε (αν το n είναι αρκετά μεγάλο, π.χ. $n \geq 10$)

$$\frac{n!}{2^n} = \frac{1\cdot2\cdot3\cdots n}{2\cdot2\cdot2\cdots2} \geq \frac{\frac{n}{2}+1}{2}\frac{\frac{n}{2}+2}{2}\cdots\frac{n}{2} \geq (n/4)^{n/2}$$

που είναι φανερό ότι τείνει στο άπειρο και μάλιστα πολύ γρήγορα.

Ένας άλλος τρόπος είναι να παρατηρήσουμε ότι για την ακολουθία $a_n = \frac{n!}{2^n}$ έχουμε $\frac{a_n}{a_{n-1}} = \frac{n}{2}$ το οποίο είναι ≥ 2 αν $n \geq 4$, άρα

$$a_n = \frac{a_n}{a_{n-1}}\frac{a_{n-1}}{a_{n-2}}\cdots\frac{a_5}{a_4}a_4 \geq \underbrace{2\cdots2}_{n-4 \text{ παράγοντες}} a_4 = 2^{n-4}a_4 \to \infty.$$

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

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

Λύση: Το αριστερό μέλος μετράει όλα τα υποσύνολα του $[n] = \{1, 2, \ldots, n\}$ μεγέθους k ενώ το δεξί μετράει τα υποσύνολα μεγέθους $n - k$. Η απεικόνιση

$$A \to [n] \setminus A$$

που απεικονίζει κάθε σύνολο στο συμπλήρωμά του είναι μια 1-1 και επί απεικόνιση από τα πρώτα σύνολα (μεγέθους k) στα δεύτερα (μεγέθους $n - k$) άρα οι δύο κατηγορίες συνόλων είναι ισοπληθικές και η ισότητα αποδείχτηκε.

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, οποιουδήποτε μεγέθους, έχει το σύνολο $\{1, 2, \ldots, n\}$. Στο δεξί μέλος ο διωνυμικός συντελεστές $\binom{n}{k}$ μετράει κι αυτός κάποιου είδους υποσύνολα.

Λύση: Στο αριστερό μέλος μετράμε όλα τα υποσύνολα του $[n]$, οποιουδήποτε μεγέθους. Στο δεξί μέλος το $\binom{n}{0}$ μετράει τα υποσύνολα μεγέθους 0, το $\binom{n}{1}$ μετράει τα υποσύνολα μεγέθους 1, κλπ, και τέλος το $\binom{n}{n}$ μετράει τα υποσύνολα μεγέθους n. Επειδή κάθε υποσύνολο έχει κάποιο μέγεθος k, με $0 \le k \le n$ το ζητούμενο έπεται.

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.

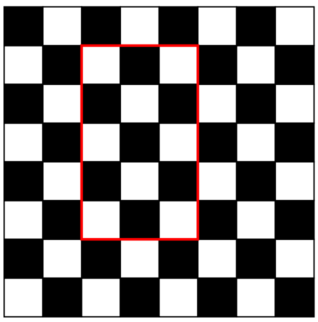
Λύση: Η απάντηση στο πρώτο είναι $\binom{10}{3}$ αφού το ότι τα τρία άτομα του προεδρείου δεν έχουν διακριτούς ρόλους είναι σα να λέμε ότι δε μας ενδιαφέρει η σειρά των επιλεγομένων.

Για το δεύτερο ερώτημα, επιλέγουμε πρώτα τον πρόεδρο (10 επιλογές), μετά τον αντιπρόεδρο (9 επιλογές) και τέλος τα τρία μέλη ταυτόχρονα ($\binom{8}{3}$ επιλογές). Σύνολο $10 \cdot 9 \cdot \binom{8}{3}$ επιλογές.

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 ($\diamondsuit$ ή $\heartsuit$) και 3 σπαθιά; Δε μας ενδιαφέρει η σειρά επιλογής των φύλλων.

Λύση: Επιλέγουμε πρώτα τα δύο κόκκινα φύλλα ($\binom{26}{2}$ επιλογές) και μετά τα 3 σπαθιά ($\binom{13}{3}$ επιλογές). Σύνολο $\binom{26}{2}\binom{13}{3}$ επιλογές.

10. Σε μια 8×8 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 8×8 σκακιέρα κι ένα από τα ορθογώνια τα οποία θέλουμε να μετρήσουμε

Δείτε στο Σχήμα.

Λύση: Ένα τέτοιο ορθογώνιο ορίζεται από την προβολή του στον x-άξονα και από την προβολή του στον y-άξονα, οι οποίες μπορούν να επιλεγούν ανεξάρτητα η μια από την άλλη. Κάθε μια από τις προβολές αυτές είναι ένα διάστημα τα άκρα του οποίου είναι δύο στοιχεία του συνόλου $\{0, 1, \ldots, 8\}$. Για να επιλέξουμε λοιπόν την κάθε προβολή έχουμε $\binom{9}{2}$ επιλογές, άρα το πλήθος των ορθογωνίων είναι $\binom{9}{2}^2$.

11. (α) Με πόσους τρόπους μπορούμε να διατάξουμε τα ψηφία $1, 2, \ldots, 9$ ώστε ανάμεσα στο 1 και το 2 να υπάρχουν ακριβώς τρία ψηφία;

(β) Με πόσους τρόπους μπορούμε να διατάξουμε τα ψηφία $1, 2, \ldots, 9$ ώστε το 1 να προηγείται του 2 και το 2 να προηγείται του 3;

Λύση: (α) Επιλέγουμε πρώτα τις θέσεις του 1 και του 2. Αυτές πρέπει να διαφέρουν κατά 3, άρα οι δυνατές θέσεις του πρώτου εκ των δύο αυτών είναι 5 (από την πρώτη έως την πέμπτη θέση). Έπειτα πρέπει να διαλέξουμε και ποιο από τα 1,2 είναι πρώτο (άλλες 2 επιλογές). Τέλος παίρνουμε μια οποιαδήποτε διάταξη των υπολοίπων 7 ψηφίων και τη βάζουμε στις υπόλοιπες θέσεις. Σύνολο $5 \cdot 2 \cdot 7!$ επιλογές.

(β) Επιλέγουμε πρώτα τη θέση της τριάδας 1,2,3 ($\binom{9}{3}$ επιλογές. Έπειτα τη σειρά των υπολοίπων ψηφίων στις υπόλοιπες θέσεις (6! επιλογές). Σύνολο $\binom{9}{3} \cdot 6!$ επιλογές.

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):
    """Ηςυνάρτησηςυπολογίζεικαιεπιστρέφειόλαταυποσύνολατουσυνόλου
              s, τοοποίομαςδίδεταιωςμιαλίστα
        , πχ.. ωςτηλίστα['a', 'b', 'c'].Τοαποτέλεσμαμαςεπιστρέφεταιωςμιαλίσταααπόσύνολαλίστα
             ( απόλίστεςδηλ.).
    """
    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], []
]
```