Test wiedzy w wersji Python (z odpowiedziami)

XV OIJ, zawody I stopnia, tura testowa
12 listopada 2020

Olimpiada Informatyczna Juniorów

Poniżej znajdują się rozwiązania zadań drugiego podejścia tury testowej zawodów I stopnia XV Olimpiady Informatycznej Juniorów (oij.edu.pl).

1. **Która z poniższych instrukcji pozwala przerwać wykonanie pętli?**

   ☐ `stop`

   ☒ `break`

   ☐ `terminate`

   ☐ `interrupt`

2. **Celem poniższego (fragmentu) programu jest obliczyć sumę liczb zapisanych w tablicy A o długości n:**

```
wynik = 0
for i in range(n):
  ???
```

   Co należy wstawić w miejsce znaków zapytania, aby program spełniał swoje założenia?

   ☐ `wynik + A[i]`

   ☒ `wynik += A[i]`

   ☒ `wynik = wynik + A[i]`

   ☐ `wynik = A[i]`

3. **Celem poniższego (fragmentu) programu jest obliczyć i wypisać n-tą liczbę Fibonacciego F_n. Zakładamy, że $F_0 = F_1 = 1$ oraz $F_n = F_{n-1} + F_{n-2}$ dla $n \geq 2$.**

```
f_aktualny, f_poprzedni = 1, 1
for i in range(2, n+1):
  f_nastepny = f_aktualny + f_poprzedni
  ???
  f_aktualny = f_nastepny
print(f_aktualny)
```

   Co należy wstawić w miejsce znaków zapytania, aby program spełniał swoje założenia?

   ☒ `f_poprzedni = f_aktualny`

   ☐ `f_aktualny = f_poprzedni`

   ☐ `f_nastepny = f_aktualny`

   ☐ `f_poprzedni = f_nastepny`

   ☐ nic nie trzeba wpisać (wystarczy zmazać znaki zapytania)

4. **Ile jest potęg dwójki o wykładniku całkowitym wśród liczb ze zbioru $\{40, 41, 42, \ldots, 590, 591\}$?**

   4

   Rozwiązanie:
   Chodziło o liczby ze zbioru $\{64, 128, 256, 512\}$.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

5. **Rozważmy fragment programu pokazany poniżej:**

```
for i in range(5):
  print('*', end='')
  j = 1
  while j < 5:
    print('*', end='')
    j *= 2
  print('*', end='')
```

Ile znaków $*$ zostanie wypisanych przez powyższy kod?

25

Rozwiązanie:
Wewnętrzna pętla wykonuje się zawsze trzy razy (dla $j \in \{1,2,4\}$), a więc każda z pięciu iteracji zewnętrznej pętli wypisuje pięć gwiazdek, czyli wypisanych gwiazdek będzie $5 \cdot 5 = 25$.

6. **Następująca funkcja powinna zwracać `true` wtedy i tylko wtedy, gdy liczba n jest pierwsza.**

```
def czy_pierwsza(n):
  i = 2
  while i*i < n:
    if n % i == 0:
      return False
    i += 1
  return True
```

Dla jakiej liczby n program zwróci nieprawidłową odpowiedź?

- ☒ 25
- ☐ 100
- ☐ 20
- ☐ 17
- ☒ 9

Rozwiązanie:
Liczby złożone n mają rozkład na czynniki (niekoniecznie pierwsze) $p,q \notin \{1,n\}$ oraz $p \cdot q = n$. Co najmniej jeden z tych czynników jest równy co najwyżej $\sqrt{n}$, bo inaczej cały iloczyn byłby większy niż n. A zatem wystarczające jest sprawdzanie podzielności przez $\{2,3,\ldots,\lfloor\sqrt{n}\rfloor\}$. W powyższym programie błąd polega na tym, że pętla wykonuje się dopóki $i^2 < n$ zamiast $i^2 \leq n$. Jeśli więc liczba n jest kwadratem liczby pierwszej (lub $n = 1$) to program nieprawidłowo zwróci `True` zamiast `False`. Tak stanie się dla liczb $3^2 = 9$ oraz $5^2 = 25$. Dla liczb 20 oraz 100 program prawidłowo stwierdzi, że liczba jest złożona (dla $i = 2$) oraz dla liczby 17 program prawidłowo stwierdzi, że liczba jest pierwsza.

7. **Które z poniższych działań obliczają ostatnią (najmniej znaczącą) cyfrę liczby n w zapisie dziesiętnym?**

- ☐ `(n // 10) * 10`
- ☒ `n % 10`
- ☐ `n // 10`
- ☒ `n - n // 10 * 10`
- ☐ `n * 10 // 100`

Rozwiązanie:
Obliczenie reszty z dzielenia przez 10 jest znanym sposobem wyłuskania ostatniej cyfry liczby. Jeśli od dzielnej odejmiemy dziesięciokrotność wyniku dzielenia przez 10 (zaokrąglonego w dół) to również otrzymamy resztę z dzielenia przez 10.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

8. **Rozważmy tabliczkę mnożenia od 1 do 10 czyli tabelę rozmiaru 10×10, w której w i-tym wierszu i j-tej kolumnie znajduje się liczba $i \cdot j$. Jaka jest suma wszystkich stu liczb zapisanych w tej tabeli?**

3025

Rozwiązanie:
Wynikiem jest $1 \cdot 1 + 1 \cdot 2 + \ldots 1 \cdot 10 + 2 \cdot 1 + 2 \cdot 2 + \ldots 2 \cdot 10 + \cdots + 10 \cdot 1 + 10 \cdot 2 + \cdots + 10 \cdot 10$. Można to zapisać następująco: $(1 + 2 + \cdots + 10) \cdot (1 + 2 + \cdots + 10)$ i obliczyć, że $1 + 2 + \cdots + 10 = 55$, a więc wynikiem jest $55 \cdot 55 = 3\,025$.

9. **Dla jakich wartości N zużycie pamięci przez tablicę zdefiniowaną jak poniżej:**

```
tab = []
for i in range(N):
  tab.append([])
  for _ in range(N):
    tab[i].append(0)
```

lub równoważnie:

```
tab = [[0] * N for _ in range(N)]
```

nie przekroczy 50 MB?

- ☐ $1\,000\,000$
- ☐ $10\,000$
- ☒ 100
- ☒ $1\,000$
- ☒ 10

Rozwiązanie:
Program tworzy N^2 zmiennych typu `int` (oraz N referencji do podtablic rozmiaru N co jest pomijalne). Jedna zmienna typu `int` zajmuje zazwyczaj około ośmiu bajtów. W połączeniu z kosztem referencji zużycie pamięci jest rzędu 9 MB dla $N = 1\,000$ oraz ponad 800 MB dla $N = 10\,000$.

10. **Rozważmy poniższą funkcję:**

```
def f(n):
  if n == 0:
    return 0
  return n % 2 + f(n // 10)
```

Funkcja ta dla liczby naturalnej n oblicza:

- ☒ liczbę cyfr nieparzystych w zapisie dziesiętnym liczby n
- ☐ parzystość przedostatniej cyfry w zapisie dziesiętnym liczby n
- ☐ liczbę jedynek w zapisie dwójkowym liczby n
- ☐ sumę cyfr w zapisie dziesiętnym liczby n
- ☐ liczbę cyfr w zapisie dwójkowym liczby n

Rozwiązanie:
Funkcja oblicza parzystość (ostatniej cyfry) liczby n (1 jeśli jest nieparzysta) oraz wynik funkcji dla n bez ostatniej cyfry, a zatem sumuje nieparzystość wszystkich cyfr liczby n.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

11. **Celem poniższego (fragmentu) programu jest sprawdzenie czy niepusty ciąg długości n zapisany w tablicy A jest posortowany niemalejąco i zapisanie wyniku w zmiennej** `posortowany`**:**

```
posortowany = True
for i in range(n-1):
  if ???:
    posortowany = False
```

Co należy wstawić w miejsce znaków zapytania, aby program spełniał swoje założenia?

- ☐ `i > i+1`
- ☒ `A[i] > A[i+1]`
- ☐ `A[i] > i+1`
- ☐ `A[i] < A[i+1]`
- ☐ `i+1 > A[i]`

12. **Rozważmy fragment poniższego programu:**

```
for i in range(1, n+1):
  for j in range(1, i+1):
    print('*', end='')
```

Ile gwiazdek mógłby (otrzymując odpowiednią wartość n) wypisać ten program?

- ☐ 20
- ☒ 28
- ☐ 12
- ☐ 7
- ☒ 10
- ☒ 6

Rozwiązanie:
Program może wypisać jedynie następujące liczby gwiazdek: $\{1, 1+2, 1+2+3, 1+2+3+4, \dots\}$, czyli liczba wypisanych gwiazdek zawsze jest liczbą *trójkątną* postaci $\frac{k(k+1)}{2}$ dla $k \in \mathbb{N}$.

13. **Które z poniższych liczb zapisanych w systemie szesnastkowym są nieparzyste?**

- ☒ 99_{16}
- ☐ 64_{16}
- ☒ AD_{16}
- ☒ CB_{16}
- ☐ DA_{16}

Rozwiązanie:
Podobnie jak w systemie dziesiętnym, każda liczba kończąca się na 0 jest parzysta, bo jest wielokrotnością 16. Zatem $99_{16} = 9 \cdot 16 + 9$ jest nieparzysta, $64_{16} = 6 \cdot 16 + 4$ jest parzysta i tak dalej. Ogólniej, wystarczy spojrzeć na parzystość ostatniej cyfry. W systemie szesnastkowym parzyste są cyfry ze zbioru $\{0, 2, 4, 6, 8, A = 10, C = 12, E = 14\}$.

14. **Silnią liczby n (zapisywaną $n!$) nazywamy iloczyn kolejnych liczb naturalnych od 1 do n włącznie. Na przykład $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$. Ile wynosi ostatnia cyfra dziesiętna sumy $1! + 2! + 3! + 4! + \cdots + 100!$?**

3

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

Rozwiązanie:

Wystarczy zsumować ostatnie cyfry liczb $1!, 2!, 3!, \ldots, 100!$. Są to kolejno: $1, 2, 6, 4, 0, 0, 0, \ldots, 0$ (ostatnia cyfra liczby $n!$ jest 0 dla wszystkich $n \geq 5$). A zatem ostatnia cyfra sumy z zadania jest ostatnią cyfrą liczby $1 + 2 + 6 + 4 = 13$ czyli 3.

15. **Ile różnych trójkątów (o dodatnim polu) można zbudować wybierając dokładnie trzy patyczki ze zbioru dziesięciu patyczków o długościach kolejno:** $1, 2, 3, \ldots, 10$**? Trójkąty uznajemy za różne jeśli zbiory użytych patyczków są różne.**

50

Rozwiązanie:

Trzy odcinki o długościach $a \leq b \leq c$ tworzą trójkąt o dodatnim polu wtedy i tylko wtedy, gdy $a + b > c$.
Jeśli $a = 1$, to nie jest możliwe, aby $a + b > c$ niezależnie od wyboru b i c. Jeśli $a = 2$, to aby utworzyć trójkąt o dodatnim polu konieczne jest, aby $c = b + 1$. Jest na to 7 możliwości: $b \in \{3, 4, 5, 6, 7, 8, 9\}$. Jeśli $a = 3$, to musi zachodzić $c \in \{b + 1, b + 2\}$. A zatem jest 11 możliwości wyboru pary (b, c): po dwie możliwości wyboru c dla $b \in \{4, 5, 6, 7, 8\}$ oraz jedna dla $b = 9$.
Analogicznie możemy stwierdzić, że dla $a = 4$ jest $3 + 3 + 3 + 2 + 1 = 12$ możliwości wyboru pary (b, c). Dla $a = 5$ jest $4 + 3 + 2 + 1 = 10$ możliwości. Dla $a = 6$ jest $3 + 2 + 1 = 6$ możliwości. Dla $a = 7$ są $2 + 1 = 3$ możliwości. Wreszcie dla $a = 8$ jest już tylko jedna możliwość tj. trójkąt $(8, 9, 10)$.
Razem trójek (a, b, c) spełniających warunki zadania jest $7 + 11 + 12 + 10 + 6 + 3 + 1 = 50$.

16. **Rozważmy poniższą funkcję:**

```
def f(x):
  return x * 2
```

Wywołanie `f(f(f(x)))` **oblicza:**

- ☒ $8x$
- ☐ $16x$
- ☐ $2x$
- ☐ $6x$
- ☐ $4x$

17. **Rozważmy fragment programu pokazany poniżej:**

```
def funkcja(n):
  wyniki = []
  # Wstaw do tablicy wyniki losowa liczbe.
  wyniki.append(random.randint(1, n))
  for i in range(1, n):
    if n % i == 0:
      wyniki.append(i)
  # Ustaw elementy znajdujace sie w wyniki
  # w losowej kolejnosci.
  random.shuffle(wyniki)
  return wyniki
```

Funkcja zwróciła następującą wartość `[35, 25, 7, 48, 1, 5]`**. Jaki był parametr** n **z jakim uruchomiono funkcję?**

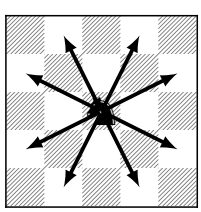
175

Rozwiązanie:

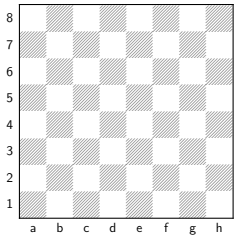
Funkcja zwraca listę wszystkich dzielników właściwych liczby n (tzn. dzielników mniejszych od n) oraz jedną losową liczbę do n. Wszystkie zwrócone liczby poza 48 są dzielnikami liczby $175 = 5^2 \cdot 7$.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

18. **Skoczek szachowy rozpoczyna podróż w polu** `a1` **i wykonuje dwa ruchy. Na ilu różnych polach może zakończyć się podróż skoczka?**



Ruchy skoczka szachowego



Pola szachownicy

10

Rozwiązanie:

Skoczek w pierwszym ruchu może się znaleźć na jednym z dwóch pól: `b3` lub `c2`. Teraz należy rozważyć zbiory pól osiągalnych w jednym ruchu z tych dwóch pól (pamiętając o polu `a1` i nie liczeniu tych samych pól dwukrotnie).

19. **Rozważmy poniższą funkcję:**

```
def wypisuj(n):
  if n < 0: return
  if n == 0:
    print('*', end='')
    return
  wypisuj(n - 1)
  wypisuj(n - 2)
```

Ta funkcja (dla odpowiednio dobranej wartości zmiennej n) może wypisać dokładnie:

- ☒ 8 gwiazdek
- ☐ 9 gwiazdek
- ☐ 6 gwiazdek
- ☒ 3 gwiazdki
- ☒ 5 gwiazdek
- ☐ 4 gwiazdki

Rozwiązanie:

Liczba wypisanych gwiazdek jest zawsze liczbą Fibonacciego.

20. **Które z wyrażeń są prawdziwe?**

- ☒ `ord('a') < ord('b')`
- ☒ `ord('B') - ord('A') == 1`
- ☐ `ord('B') > ord('C')`
- ☒ `ord('A') < 200`

Rozwiązanie:

W zadaniu należało pamiętać najbardziej podstawowe właściwości tablicy ASCII. Kolejne małe litery mają przypisane kolejne kody ASCII (od 97 do 122, ale nie trzeba było tego pamiętać dokładnie) oraz kolejne wielkie litery mają przypisane kolejne kody ASCII (od 65 do 90).

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

21. **Rozważmy poniższą funkcję:**

```
def szyfruj(s):
  wynik = ''
  for i in range(len(s)):
    if s[i] == 'z':
      wynik += 'a'
    else:
      wynik += chr(ord(s[i]) + 1)
  return wynik
```

Dla jakiego napisu funkcja zwróci `nbsjb`**?**

maria

Rozwiązanie:
Powyższy program jest implementacją szyfru Cezara z przesunięciem o 1. Każdy znak wejścia jest przesuwany o jedną literę alfabetu do przodu (litera z jest przesuwana do a). Aby więc cofnąć ten proces, wystarczy przesunąć każdą literę zaszyfrowanego napisu do poprzedniej litery w alfabecie.

22. **Rozważmy poniższą funkcję:**

```
def funkcja(s):
  elementy = set()
  for x in s:
    if (x >= 'a') and (x <= 'z'):
      elementy.add(ord(x) - ord('a'))
  return len(elementy)
```

Dla jakich napisów funkcja zwróci wartość 3**?**

- ☐ `olimpiada`
- ☒ `Bajt`
- ☒ `kajak`
- ☐ `aaa`
- ☒ `oijoij`
- ☐ `XYZ`

Rozwiązanie:
Powyższy program oblicza liczbę różnych (`set` usuwa powtórzenia) małych liter (pozostałe znaki ignorujemy) w napisie s. Do zbioru `elementy` dodawane są pozycje liter w alfabecie indeksowanym od 0 (tzn. znak `a` skutkuje wstawieniem 0 do zbioru, znak `b` wstawieniem 1 itd.).

23. **Rozważmy poniższą funkcję:**

```
def funkcja(a, b):
  if b == 0: return a
  return funkcja(b, a % b)
```

Co zwraca powyższa funkcja?

- ☐ a^b
- ☐ $a \cdot b$
- ☒ $\mathrm{nwd}(a, b)$
- ☐ $\mathrm{nww}(a, b)$
- ☐ $a + b$

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

Rozwiązanie:
Powyższa funkcja jest implementacją algorytmu Euklidesa do obliczania największego wspólnego dzielnika dwóch liczb.

24. **Ile liczb ze zbioru $\{92, 93, 94, \ldots, 293\}$ jest podzielnych przez 5, ale niepodzielnych przez 7?**

34

Rozwiązanie:
Liczbę, która spełnia warunek podzielności przez 5 oraz niepodzielności przez 7 nazwijmy *fajną*. Obliczmy ile jest liczb fajnych w zbiorze $\{1, 2, \ldots, 293\}$, a następnie odejmijmy od tego ile jest liczb fajnych w zbiorze $\{1, 2, \ldots, 91\}$.
Aby obliczyć ile jest liczb fajnych w zbiorze $\{1, 2, \ldots, n\}$ zauważamy, że liczb podzielnych przez 5 jest $\lfloor \frac{n}{5} \rfloor$. Od tego należy odjąć liczby podzielne przez $35 = 5 \cdot 7$, których jest $\lfloor \frac{n}{35} \rfloor$.
A zatem, wynikiem zadania jest:

$$\lfloor \frac{293}{5} \rfloor - \lfloor \frac{293}{35} \rfloor - \left(\lfloor \frac{91}{5} \rfloor - \lfloor \frac{91}{35} \rfloor \right) = 58 - 8 - (18 - 2) = 34$$

25. **Rozważmy poniższą funkcję:**

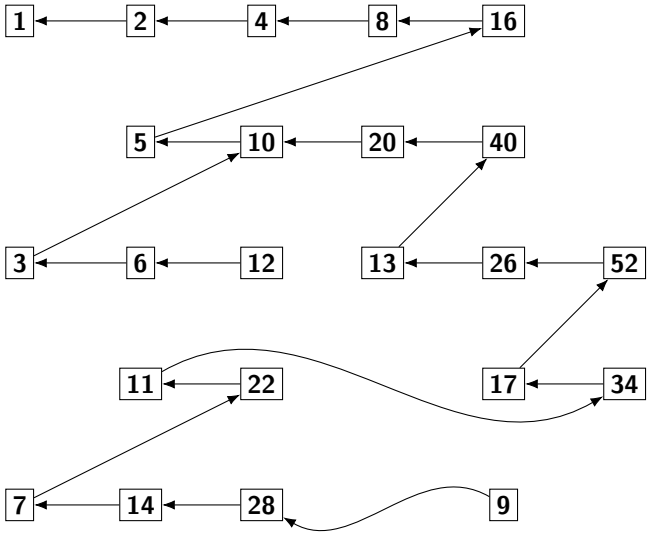
```
def probuj(n):
  print('*', end='')
  if n == 1: return
  if n % 2 == 0:
    probuj(n // 2)
  else:
    probuj(3 * n + 1)
```

Jaka jest najmniejsza dodatnia naturalna liczba n, dla której wywołanie `probuj(n)` spowoduje wypisanie dokładnie dziewięciu gwiazdek?

6

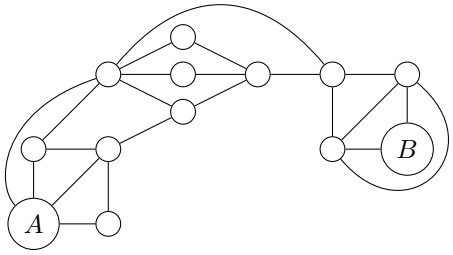
Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

Rozwiązanie:
Narysujmy punkty odpowiadające kolejnym liczbom naturalnym i połączmy je strzałkami – z liczby x rysujemy strzałkę do $3x+1$ jeśli x jest nieparzyste, a do $\frac{x}{2}$, jeśli x jest parzyste. Innymi słowym, rysujemy graf skierowany, dla którego każda krawędź prowadzi od pewnej liczby x do następnej, którą będzie rozważał nasz program. Szukamy wierzchołka grafu, dla którego ścieżka do 1 będzie miała dokładnie 8 krawędzi (czyli 9 wierzchołków).
Jeśli zaczniemy budować ten graf tworząc wierzchołki i krawędzie dla kolejnych liczb naturalnych od 1 wzwyż otrzymamy w pewnym momencie następujący rysunek:



Ścieżka od wierzchołka 6 do wierzchołka 1 ma 9 wierzchołków. Ścieżki od pozostałych wierzchołków ze zbioru $\{1,2,\ldots,5\}$ mają inną długość. Stąd odpowiedzią, której należy udzielić jest 6. Ogólnie, liczby ze zbioru $\{2,3,4,5,6\}$ leżą za to na wspólnej ścieżce, na której dziewiątym wierzchołkiem jest 6.
Zadanie odnosi się do tak zwanej hipotezy Collatza `https://pl.wikipedia.org/wiki/Problem_Collatza`, która mówi, że dla każdego początkowego n zawsze dojdziemy do 1. Jak dotąd, nikomu nie udało się ani tego udowodnić, ani też znaleźć przykładu, dla którego tak nie jest.

26. **Ile najmniej krawędzi można usunąć z grafu z rysunku, aby nie istniała ścieżka między wierzchołkami A i B?**

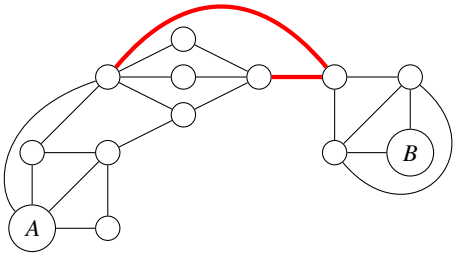


Graf

2

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

Rozwiązanie:
Sytuację obrazuje poniższy rysunek:



27. **Rozważmy fragment programu pokazany poniżej:**

```
ile = 0
liczby = [4, 5, 2, 4, 9, 3, 1, 5, 0, 1]
for i in range(len(liczby)):
  for j in range(i+1, len(liczby)):
    for k in range(j+1, len(liczby)):
      if liczby[i] + liczby[j] + liczby[k] == 10:
        ile += 1
print(ile)
```

Jaką liczbę wypisze na wyjście ten program?

14

Rozwiązanie:
Powyższy program znajduje liczbę trójek pozycji (i, j, k) $(i < j < k)$ w tablicy `liczby`, że suma liczb na tych pozycjach jest równa 10. Wynik pozostanie taki sam po posortowaniu tablicy: $[0, 1, 1, 2, 3, 4, 4, 5, 5, 9]$. Są dwie trójki (i, j, k) dające sumę 10 dzięki elementom o wartościach $0, 1, 9$, jedna trójka (i, j, k) dająca sumę 10 dzięki elementom o wartościach $0, 5, 5$, osiem trójek (i, j, k) dających sumę 10 dzięki elementom o wartościach $1, 4, 5$, dwie trójki (i, j, k) dające sumę 10 dzięki elementom o wartościach $2, 3, 5$ oraz jedna trójka (i, j, k) dająca sumę 10 dzięki elementom o wartościach $2, 4, 4$.

28. **Rozważmy program, który wczytuje liczbę naturalną n i wykonuje potem dokładnie $n\sqrt{n}$ operacji elementarnych w celu obliczenia wyniku. Uruchamiono program na komputerze, który może wykonać 10^8 operacji elementarnych w ciągu sekundy. Dla jakiej wartości n wykonanie programu zajmie około dwie sekundy?**

- ☐ $1\,000\,000$
- ☐ $1\,500$
- ☒ $350\,000$
- ☐ $4\,000$
- ☐ 200

Rozwiązanie:
Mamy do rozwiązania równanie $n\sqrt{n} = 2 \cdot 10^8$. Po podniesieniu do kwadratu otrzymujemy $n^3 = 4 \cdot 10^{16}$, a zatem $n = \sqrt[3]{4 \cdot 10^{16}}$ czyli więcej niż $10^5 = 100\,000$ oraz mniej niż $10^6 = 1\,000\,000$. Właściwe rozwiązanie równania to $n \approx 341\,995$.
Program dla $n = 350\,000$ będzie wykonywał się ok. 2.07 sekundy.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

29. **Mówimy, że słowo S zawiera podciąg T, jeśli możliwe jest zakrycie niektórych liter w S, aby odczytując pozostałe litery od lewej do prawej uzyskać T. Na przykład: słowo `anas` jest (a słowo `ansa` nie jest) podciągiem słowa `ananas`. Rozważmy wszystkie słowa o długości 6 złożone jedynie z liter `a`, `b`, `c`. Ile spośród tych słów zawiera w sobie (co najmniej raz) podciąg `abc`?**

233

Rozwiązanie:

(Sposób 1) Niech $R_{n,m}$ oznacza liczbę słów n-literowych zawierających jako podciąg m pierwszych liter słowa `abc`.
Zauważmy, że dla $m \in \{0,1,2\}$ zachodzi: $R_{n,m} = 2 \cdot R_{n-1,m} + R_{n-1,m-1}$ (albo do słowa doklejamy kolejną literę pasującą do kolejnej litery podciągu `abc`, albo doklejamy inną literę, czego możemy dokonać na dwa sposoby). Ponadto $R_{n,3} = 3 \cdot R_{n-1,3} + R_{n-1,2}$ (albo słowo $n-1$ literowe miało już w sobie podciąg `abc` i można było do niego dopisać dowolną literę albo miało podciąg `ab` i konieczne było dopisanie litery `c`).
Możemy na podstawie powyższych rozważań zrobić tabelę z wartościami $R_{n,m}$:

	$m=0$	$m=1$	$m=2$	$m=3$
$n=0$	1			
$n=1$	2	1		
$n=2$	4	4	1	
$n=3$	8	12	6	1
$n=4$	16	32	24	9
$n=5$	32	80	80	51
$n=6$	?	?	?	233

Pozostałe wyniki dla $n = 6$ nie były nam potrzebne, więc zostały pominięte. Ostatecznie wynikiem zadania jest $R_{6,3} = 233$.
(Sposób 2) Zgadnijmy pozycję p (indeksujemy pozycje od 1 do 6) pierwszej litery `c` występującej po podciągu `ab` w naszym słowie.
Jeśli $p = 3$, to słowo musi wyglądać następująco: `abc???`, gdzie `?` oznacza dowolny znak ze zbioru $\{$`a`,`b`,`c`$\}$ (27 możliwości).
Jeśli $p = 4$, to wybieramy wśród pierwszych trzech pozycji dwie pozycje. Na pierwszej wybranej postawimy `a`, a na drugiej postawimy `b`. Pozycja nie wybrana (wśród pierwszych trzech) może być uzupełniona na dwa sposoby niezależnie od przypadku. Ostatnie dwa znaki mogą być uzupełnione dowolnie, a zatem mamy: $\binom{3}{2} \cdot 2 \cdot 3^2 = 3 \cdot 2 \cdot 9 = 54$ możliwości.
Jeśli $p = 5$, to analogicznie wśród pierwszych czterech pozycji wybieramy pozycje x gdzie pojawi się pierwsze wystąpienie `a` oraz $y > x$ gdzie pojawi się pierwsze wystąpienie `b` za wystąpieniem `a`. Każdy z pozostałych dwóch znaków wśród pierwszych czterech można wybrać na dwa sposoby. Ostatni znak może być uzupełniony dowolnie, a zatem mamy: $\binom{4}{2} \cdot 2^2 \cdot 3 = 6 \cdot 4 \cdot 3 = 72$ możliwości.
Analogicznie, jeśli $p = 6$, mamy $\binom{5}{2} \cdot 2^3 = 10 \cdot 8 = 80$ możliwości.
Razem otrzymujemy $27 + 54 + 72 + 80 = 233$ słów spełniających warunki zadania.

30. **Pewien program wykonuje $\frac{n}{1} + \frac{n}{2} + \cdots + \frac{n}{n}$ operacji dla danych wejściowych rozmiaru n. Jaką złożoność obliczeniową ma ten program?**

- ☐ $\Theta(\log n)$
- ☐ $\Theta(n\sqrt{n})$
- ☐ $\Theta(n)$
- ☐ $\Theta(n^2)$
- ☒ $\Theta(n \log n)$

Rozwiązanie:

Mianownik każdego ułamka możemy zaokrąglić w górę lub w dół do najbliższej potęgi dwójki, aby otrzymać:

$$\frac{n}{1} + \frac{n}{2} + \frac{n}{4} + \frac{n}{4} + \frac{n}{8} + \cdots \le \frac{n}{1} + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} + \frac{n}{5} + \cdots + \frac{n}{n} \le \frac{n}{1} + \frac{n}{2} + \frac{n}{2} + \frac{n}{4} + \frac{n}{4} + \ldots$$

Obie sumy zaokrąglonych ułamków są rzędu $\Theta(n \log n)$, bo składają się z $\Theta(\log n)$ grup ułamków o tej samej wartości i sumie każdej grupy odpowiednio: $\frac{n}{2}$ w lewej sumie oraz n w prawej. Nie ma tutaj znaczenia jak duża jest wartość ostatniej, być może niepełnej grupa (wiadomo, że jest rzędu $O(n)$).

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".