

Test wiedzy algorytmicznej – rozwiązania

XXI OIJ, zawody I stopnia
22 października 2026



1. W pewnym dawnym komunikatorze używane było nietypowe kodowanie – każda cyfra była zapisywana jako dwie litery w następujący sposób:

AD	0		FD	5
BD	1		GD	6
CD	2		HD	7
DD	3		ID	8
ED	4		JD	9

Jaka liczba została zakodowana jako CDADFDJDBD?

- ☐ A. 1234
☐ B. 93014
☐ C. 43210
☒ D. 20591
2. Rozważmy poniższą funkcję:

wersja C++

```
int f(int x, int y, int z)
{
    if (x > y)
        return z+1;
    else
        return z-1;
}
```

wersja Python

```
def f(x, y, z):
    if x > y:
        return z+1
    else:
        return z-1
```

Jaki będzie wynik działania $f(8,11,0)+f(9,3,5)+f(7,7,10)$?

14

Rozwiązanie: $f(8,11,0) = -1$, $f(9,3,5) = 6$, $f(7,7,10) = 9$.



3. Początkowa wartość zmiennej a jest z zakresu od 0 do 9. **Określ, dla których zestawów instrukcji zmienna a zawsze na końcu przyjmie wartość 5.**

wersja C++

- ☐ **A.** `if (a < 2)`
 `a = 5;`
 `if (a >= 2)`
 `a = 9;`
- ☐ **B.** `if (a < 2)`
 `a = 1;`
 `else`
 `a = 9;`
- ☒ **C.** `if (a > 8)`
 `a = 1;`
 `if (a <= 8)`
 `a = 5;`
- ☐ **D.** `if (a > 8)`
 `a = 1;`
 `else`
 `a = 5;`

wersja Python

- ☐ **A.** `if a < 2:`
 `a = 5`
 `if a >= 2:`
 `a = 9`
- ☐ **B.** `if a < 2:`
 `a = 1`
 `else:`
 `a = 9`
- ☒ **C.** `if a > 8:`
 `a = 1`
 `if a <= 8:`
 `a = 5`
- ☐ **D.** `if a > 8:`
 `a = 1`
 `else:`
 `a = 5`

Rozwiązanie: W wariancie **C** pierwsza instrukcja zmienia wartość $a = 9$ na $a = 1$, inne pozostawiając bez zmian. Druga instrukcja zatem zawsze napotka $a \leq 8$ i zmieni je na 5. W pozostałych wariantach można na przykład zauważyć, że zaczynając od $a = 9$ nie dostaniemy 5.

4. Zapis $1AF_{16}$ w systemie szesnastkowym oznacza liczbę:

- ☒ **A.** 000110101111_2
- ☒ **B.** 431_{10}
- ☐ **C.** 2026_{10}
- ☒ **D.** 012233_4

Rozwiązanie: Wygodnie jest skorzystać z faktu, że jednej cyfrze szesnastkowej odpowiadają 4 cyfry dwójkowe (np. $A_{16} = 1010_2$), a także dwie cyfry czwórkowe (np. $A_{16} = 22_4$).

5. Rozważmy poniższą funkcję:

wersja C++

```
int f(int x)
{
    if (x%7 == 2)
        return 1;
    else
        return 0;
}
```

wersja Python

```
def f(x):
    if x%7 == 2:
        return 1
    else:
        return 0
```

Jaki będzie wynik działania $f(1)+f(2)+\dots+f(50)$?

7

Rozwiązanie: Wartość funkcji jest równa 1 dla liczb dających resztę 2 z dzielenia przez 7, czyli 2, 9, 16, ..., 42. Takich liczb jest 7.

6. Automatyczne urządzenie pakujące otrzymuje kolejno kartony o wagach 4 kg, 2 kg, 3 kg, 5 kg, 1 kg, 4 kg, 2 kg. Kartony są ładowane do pojemników o maksymalnej ładowności 7 kg. Nie wolno zmieniać kolejności kartonów, a w jednym pojemniku muszą znajdować się kolejne kartony z taśmy. **Jaka jest najmniejsza liczba potrzebnych pojemników?**

4

Rozwiązanie: W pojemniku zmieszczą się pierwsze dwa kartony (o łącznej wadze $4+2 = 6$ kg), ale trzeci nie wejdzie razem z czwartym. Zatem trzeci karton (3 kg) będzie w pojemniku sam, czwarty razem z piątym ($5 \text{ kg} + 1 \text{ kg}$), a szósty z siódmym ($4 \text{ kg} + 2 \text{ kg}$).

7. Rozważmy poniższą funkcję:

wersja C++

```
bool f(string s)
{
    int n = s.length();
    for(int i = 0; i < n-3; i++)
        if (s[i] != s[i+3])
            return false;
    return true;
}
```

wersja Python

```
def f(s):
    n = len(s)
    for i in range(0, n-3):
        if s[i] != s[i+3]:
            return False
    return True
```

Zaznacz wszystkie ciągi znaków, dla których funkcja zwróci true/True.

- ☒ A. abcbabc
☒ B. aaaaaaaa
☐ C. ababab
☒ D. xxyxyx

Rozwiązanie: Dla każdego i porównujemy literę i z literą $i + 3$. Dla ciągu znaków ababab nie zgadza się już dla $i = 0$.

8. Elementy 1,2,3 można ustawić w ciąg na 6 różnych sposobów: (1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1).

Na ile różnych sposobów można ustawić w ciąg elementy 1,2,2,3? Dla elementów o równych wartościach, wybór tych samych pozycji, ale w różnej kolejności liczymy tylko raz.

12

Rozwiązanie: Sprytną metodą jest zauważenie, że gdybyśmy mieli cztery różne elementy (1,2,3,4), moglibyśmy je ustawić na 24 sposoby ($4 \cdot 3 \cdot 2 \cdot 1$). Jeśli teraz zamienimy 4 na drugą kopię elementu 2, dwa ustawienia „skleją się” w jedno (np. (1,2,3,4) i (1,4,3,2) zmienią się na (1,2,3,2)). Zatem teraz jest 12 różnych ustawień.

9. Rozważmy poniższy fragment kodu:

wersja C++

```
for(int i = 0; i < 100; i++)
    for(int j = 0; j < 100; j++)
        if (i == j+1 || j == i+1)
            cout << "!" << endl;
```

wersja Python

```
for i in range(100):
    for j in range(100):
        if i == j+1 or j == i+1:
            print("!")
```

Podaj, ile razy na wyjście wypisze się znak wykrzyknika "!"

198

Rozwiązanie: Znak wykrzyknika wypisuje się, gdy zmienne i , j różnią się dokładnie o 1: (0,1), (1,2), ..., (98,99), (1,0), (2,1), ..., (99,98). Takich par jest $2 \cdot 99 = 198$.

10. Tablica zadeklarowana poniżej może posłużyć do wygenerowania napisu "oij" poprzez zastosowanie odpowiednich indeksów w odwołaniach do jej elementów.

wersja C++

wersja Python

```
string t[4]={"ajko","nelki","wejo","rosi"};    t = ["ajko","nelki","wejo","rosi"]
```

Które z poniższych odwołań dają taką możliwość?

- ☒ A. `t[2][3]`, `t[3][3]`, `t[2][2]`
- ☐ B. `t[3][4]`, `t[4][4]`, `t[1][2]`
- ☐ C. `t[0][5]`, `t[2][5]`, `t[0][0]`
- ☒ D. `t[0][3]`, `t[1][4]`, `t[2][2]`
11. Pewną ankietę wypełniło 1000 osób. Ankieta składała się z pewnej liczby pytań, w których można było odpowiedzieć „zgadzam się”, „nie zgadzam się” lub „nie mam zdania”. Po badaniu okazało się, że każda osoba wypełniła ankietę inaczej – nie było dwóch takich samych zestawów odpowiedzi. **Ile minimalnie musiało być pytań w ankiecie, żeby taka sytuacja była możliwa?**

7

Rozwiązanie: Jedno pytanie daje 3 możliwości odpowiedzi, dwa pytania – 9 możliwości, trzy pytania – $3 \cdot 3 \cdot 3 = 27$ możliwości, i ogólnie k pytań daje 3^k możliwości. Szukamy najmniejszego k , dla którego $3^k \geq 1000$. Właściwą wartością jest 7 ($k = 6$ jest jeszcze za małe: $3^6 = 729$, $3^7 = 2187$.)



12. Organizator dysponuje pewną liczbą sal, w których odbywają się próby. Zaplanowano sześć prób w następujących przedziałach czasu:

9:00 - 12:00

10:00 - 13:00

12:00 - 15:00

13:00 - 16:00

14:00 - 17:00

16:00 - 18:00

Próba kończąca się w chwili t zwalnia salę tak, że kolejna próba może zacząć się w tej sali w tej samej chwili. Ile co najmniej sal potrzeba, aby wszystkie próby mogły odbyć się zgodnie z planem?

3



Rozwiązanie: Z powyższego rysunku widać, że 3 sale wystarczą. O 14:30 trzy próby odbywają się naraz, więc lepiej się nie da.

13. Rozważmy poniższą funkcję:

wersja C++

```
int f(int x)
{
    if (x < 2)
        return 1;
    else
        return x * f(x-2);
}
```

wersja Python

```
def f(x):
    if x < 2:
        return 1
    else:
        return x * f(x-2)
```

Jaki będzie wynik wywołania $f(8)$?

384

Rozwiązanie: $f(8) = 8 * f(6) = 8 * 6 * f(4) = 8 * 6 * 4 * f(2) = 8 * 6 * 4 * 2 * f(0) = 8 * 6 * 4 * 2 * 1 = 384$.

14. Na ekranie wyświetlono numery w kolejności

4, 1, 3, 2, 5.

Możesz w jednym ruchu zamienić miejscami tylko dwa sąsiednie numery. **Ile co najmniej ruchów potrzebujesz, aby numery znalazły się w kolejności rosnącej?**

4

Rozwiązanie: Zamieniamy kolejno $4 \leftrightarrow 1$, $4 \leftrightarrow 3$, $4 \leftrightarrow 2$, $3 \leftrightarrow 2$. Zauważmy, że mniej zamian nie wystarczy: jeśli większa liczba stała na początku przed mniejszą (np. 4 przed 2), to na pewno będą musiały zostać zamienione. Nasz ciąg ruchów zamienia wyłącznie większe liczby stojące przed mniejszymi.

15. Do magazynu przyjeżdżają kolejno paczki oznaczone numerami 1, 2, 3, 4, 5. Magazynier może położyć kolejną przyjeżdżającą paczkę na stosie albo zabrać paczkę ze szczytu stosu i wysłać ją do odbiorcy. Nie może wyciągać paczek ze środka stosu. **Które z poniższych kolejności wysyłania wszystkich paczek są możliwe?**

- ☒ A. 3, 2, 1, 5, 4
☐ B. 3, 1, 2, 5, 4
☒ C. 2, 4, 3, 1, 5
☐ D. 4, 3, 5, 1, 2

Rozwiązanie: W wariancie A wrzucamy na stos 1, 2, 3, potem ściągamy kolejno 3, 2, 1, wrzucamy 4, 5, ściągamy 5 i na końcu 4. W wariancie C wrzucamy 1 i 2, ściągamy 2, wrzucamy 3 i 4, ściągamy 4, potem 3 i 1. Na końcu od razu wysyłamy 5.

16. Ile znaków + wypisze poniższy fragment kodu?

wersja C++

```
long long a = 2;
while (a < 1000000)
{
    a = a*a;
    cout << "+" << endl;
}
```

wersja Python

```
a = 2
while a < 1000000:
    a = a*a
    print("+")
```

5

Rozwiązanie: Liczba znaków + odpowiada okrężeniom pętli. W kolejnych okrężeniach wartość a będzie równa kolejno 2, 4, 16, 256, 65536 i w następnym przekroczy 1 000 000.

17. W rzędzie ustawiono osiem żarówek. Każda z nich może być zapalona (1) lub zgaszona (0). Początkowy stan żarówek jest następujący

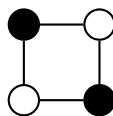
1 1 1 0 1 1 1 0.

Naciśnięcie przycisku dowolnej żarówki zmienia jej stan oraz stan żarówki znajdującej się bezpośrednio po prawej stronie (jeśli taka istnieje). Zmiana stanu oznacza, że zapalona żarówka gaśnie, a zgaszona się zapala. **Ile co najmniej razy należy nacisnąć przyciski, aby wszystkie żarówki były zgaszone?**

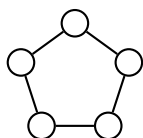
4

Rozwiązanie: Pierwszy przycisk musi być naciśnięty, bo w żaden inny sposób nie uda się zgasić pierwszej żarówki. To zgasi również drugą żarówkę, i ponieważ nie chcemy jej zapalać ponownie, nie naciskamy drugiego przycisku. Rozumując dalej podobnie, musimy nacisnąć trzeci przycisk, potem czwarty, pominąć piąty i w końcu nacisnąć szósty, co zgasi wszystkie żarówki.

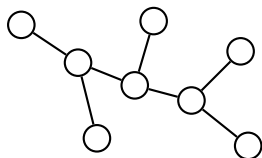
18. Na podanych obrazkach chcemy pokolorować niektóre kropki na czarno tak, żeby każde dwie kropki połączone kreską miały różne kolory, jak na poniższym przykładzie:



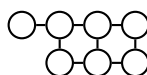
Dla których obrazków jest to możliwe?



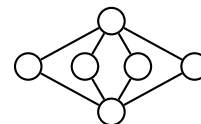
A.



B.

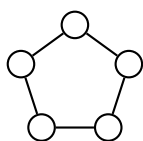


C.

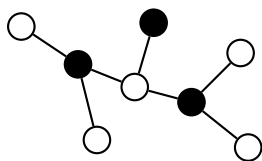


D.

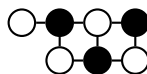
Rozwiązanie: Dla obrazka **A** jest to niemożliwe, zawsze któreś dwa sąsiednie wierzchołki muszą mieć taki sam kolor. Rozwiązania dla pozostałych obrazków podane są poniżej.



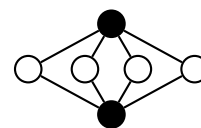
A.



B.



C.



D.

19. Terminal wyświetla jedną cyfrę od 0 do 9. Początkowo wyświetla 0. Ma dwa przyciski:

- przycisk A: zwiększa wyświetlaną liczbę o 3, a jeśli wynik przekracza 9, pozostawia jego ostatnią cyfrę,
- przycisk B: podwaja wyświetlaną liczbę, również pozostawiając tylko ostatnią cyfrę wyniku.

Ile co najmniej naciśnieć potrzeba, aby uzyskać na wyświetlaczu cyfrę 7?

5

Rozwiązanie: W jednym naciśnięciu można otrzymać 3 (i tylko 3). Drugie naciśnięcie (dowolnego przycisku) zawsze da 6. Trzy naciśnięcia pozwolą uzyskać 9 (przyciskiem A) lub 2 (przyciskiem B). W czwartym kroku z nowych liczb może pojawić się 8 (z 9 przyciskiem B), 5 (z 2 przyciskiem A) lub 4 (z 2 przyciskiem B). Dopiero piąte naciśnięcie pozwoli na otrzymanie 7, jeśli mając na wyświetlaczu 4 naciśniemy przycisk A.

20. Rozważmy poniższą funkcję:

wersja C++

```
void f(int n)
{
    if (n == 0)
        return;
    for(int i = 0; i < n; i++)
        cout << "x" << endl;
    f(n/2);
    f(n/2);
}
```

wersja Python

```
def f(n):
    if n == 0:
        return
    for i in range(n):
        print("x")
    f(n//2)
    f(n//2)
```

Ile znaków wypisze się po wywołaniu `f(16)`?

80

Rozwiązanie: Wywołanie `f(16)` wypisze 16 znaków, a potem dwukrotnie wywoła `f(8)`. Każde z dwóch wywołań `f(8)` wypisze po 8 znaków (czyli łącznie znowu 16) i wygeneruje dwa wywołania `f(4)`. Zatem będą łącznie cztery wywołania `f(4)`, które razem wypiszą znowu 16 znaków, i kontynuując to rozumowanie dostaniemy osiem wywołań `f(2)` oraz 16 wywołań `f(1)`. Łącznie wypisze się $5 \cdot 16 = 80$ znaków.