

# Czy dodzwoniłem się do internetu? - zadania kwalifikacyjne

Jacek Kurek, Michał Radwański

Marzec 2020

## 1 Informacje wstępne

1. Rozwiązania wysyłamy na adres `jacek.kurek21+www2020@gmail.com`
2. Rozwiązania najlepiej wysłać w postaci archiwum (w ludzkim formacie) zawierającego pliki `.py` oraz `pdf` z rozwiązaniami zadań tekstowych.
3. Pliki `.py` powinny poprawnie uruchamiać się w pythonie w wersji 3.6 lub wyższej, oraz nie korzystać z zewnętrznych bibliotek poza `numpy`. Warto załączyć instrukcję obsługi. W razie wątpliwości, określ użytą wersję Pythona.
4. Pytania i wątpliwości proszę kierować na `jacek.kurek21+www2020@gmail.com`
5. Rozwiązania części zadań łatwo znaleźć w internecie, prosimy jednak o rozwiązywanie ich samodzielnie
6. Za pozostawione w kodzie klucze prywatne, sekrety czy hasła do konta email organizatorzy warsztatów nie odpowiadają.
7. Wszelki udostępniony kod źródłowy jest na licencji, która umożliwia nam swobodny dostęp, modyfikację i uruchomienie, a także dalsze sublicencjonowanie.

## 2 Python

Prosimy, aby rozwiązać zadania z Pythona krok po kroku, bez omijania w sprytny sposób zapisów treści zadań (czyli np. jeśli prosimy o funkcję `funkcja`, to nie wolno `from magic_module import funkcja`).

Niezależna komisja ekspertów oceni (mając na uwadze, że jest to problem nierozstrzygalny) czy w rozwiązaniu użyte zostały już zaimplementowane do tego celu metody języka Python.

### 2.1 BMP

Chcemy, abyście napisali enkoder i dekoder obrazków BMP. Format powstał ewolucyjnie w wyniku potrzeb Microsoftu na prostą metodę serializacji obrazów. Specyfikacja formatu jest dostępna gdzieś głęboko na stronie Microsoftu, natomiast polecamy opis na **angielskiej Wikipedii**<sup>1</sup>.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/BMP\\_file\\_format](https://en.wikipedia.org/wiki/BMP_file_format)

### 2.1.1 Enkoder (10 pkt)

Pod [linkiem](#)<sup>2</sup> znajdują się pliki \*.`numpy`. Każdy z nich zawiera macierz pikseli, które mają się znaleźć w obrazku. Aby z poziomu Pythona otworzyć taki plik, można użyć `numpy`:

```
import numpy
matrix = numpy.load('ścieżka/do/pliku.numpy')
```

Napisz program, który wczyta plik `.numpy`, a następnie wyprodukuje poprawny plik BMP, który zawiera dokładnie takie same dane. Nie musisz się przejmować kompresją — wystarczy poprawny nieskompresowany obrazek. Pomyśl w jaki sposób zaprojektować program, tak aby można było w przyszłości rozszerzyć go o możliwość zapisu danych w innych formatach (przykładowo, format BMP zezwala na osadzenie obrazka PNG).

### 2.1.2 Dekoder (10 pkt)

Ten program ma pełnić odwrotną rolę — zamieniać plik BMP na plik `.numpy`. Aby nie narzucać zbyt wiele pracy, program powinien działać poprawnie dla obrazków z linku. Obrazki te nie posiadają skompresowanych danych. Program zakodowany enkoderem powinno się móc zdekodować dekodorem, i na odwrót. W obu przypadkach bitmapa powinna nie być naruszona. Zapis macierzy `numpy` do pliku robi się poprzez `numpy.save('plik.numpy', matrix)`.

## 2.2 Jak skutecznie RFC (15 pkt)

Dokumenty RFC opisują szczegółowo sposób działania wielu protokołów komunikacji jak i formatów. Znajdź RFC3629 opisujący kodowanie UTF-8, i na jego podstawie zaimplementuj kodowanie oraz dekodowanie znaków w UTF-8 — w przypadku kodowania, załóż że masz dany kod znaku, a chcesz uzyskać ciąg bajtów, i na odwrót. Dla przykładu, dla kodu 252, poprawnym wynikiem jest `b'\xc3\xbc'`, który odpowiada znakowi ü (ale tej ostatniej konwersji robić nie trzeba).

## 3 Transmisja danych

### 3.1 Szum w kanale (5 pkt)

Kanałem wysłałem pewną wiadomość, bit po bicie. Niestety, kanał jest wadliwy i każdy bit ma 1% szansy na zmianę wartości z 0 na 1 i vice versa. Po ilu wysłanych bitach mam co najmniej 20% szans, że wiadomość uległa przekłamaniu?

### 3.2 Ochrona przed błędami (5 pkt)

Zapoznaj się z pojęciem Kodów Hamminga. Załóżmy, że wysyłamy wiadomość jak w poprzednim zadaniu, tym razem jednak przed wysłaniem kodujemy ją używając kodowania Hamminga. Ile bitów mogę wysłać, zanim prawdopodobieństwo tego że wiadomość uległa uszkodzeniu (nie da się jej odtworzyć) wynosi co najmniej 20%?

### 3.3 Kodowanie wieloma symbolami (5 pkt)

Oprócz kodowania binarnego, w różnych zastosowaniach pojawiają się kodowania używające więcej niż dwóch znaków. Np. system dziesiętkowy którego używamy na co dzień używa 10 symboli.

W wypadku elektroniki, różne symbole odpowiadają np. pośrednim poziomom napięcia - oprócz niskiego i wysokiego, rozróżniamy wtedy napięcia pośrednie.

<sup>2</sup><http://students.mimuw.edu.pl/~mr395415/www16/modemwww/>

Załóżmy więc, że zamiast wysyłać bity, wysyłamy w każdej sekundzie po jednym z  $n$  symboli. Wiadomość binarną możemy w jakiś sposób zakodować na tych  $n$  symbolach, wysłać i odkodować. Jak zwiększyło się tempo transmisji, zakładając, że niezależnie od  $n$  jesteśmy w stanie wysłać symbole w tym samym tempie?

## 4 Ustalanie transmisji

### 4.1 Rozmowa we dwoje (5 pkt)

Romeo i Julia udali się na randkę. Młodzi kochankowie są bardzo nieśmiali i musi minąć dłuższa chwila, żeby któreś z nich odważyło się odezwać. Niestety, często zdarza im się odezwać w tym samym momencie, po czym zawstydzeni milkną. Na następny raz zdecydowali się znaleźć sposób na rozwiązanie tego problemu. Ustalają więc wspólny protokół, który będą stosować w sytuacji, kiedy oboje będą próbować się odezwać na raz.

Zaprojektuj taki protokół. Pamiętaj, że mamy XXIw. i równouprawnienie, więc nie powinien on faworyzować żadnej ze stron rozmowy.

### 4.2 Cisza czy zera? (10 pkt)

Trwa ogólnopolska kwarantanna więc staracie się nie wychodzić z domu. Umówiliście się z kolegą z naprzeciwka, że będziecie komunikować się za pomocą błysków latarki. Jako że aspirujecie do miana młodych informatyków, wiadomości wysyłacie będziecie zakodowane w systemie binarnym - światło oznacza jedynekę, brak światła oznacza zero.

Pojawił się jednak problem, jak rozróżnić sytuacje w której kolega przestał nadawać / upadła mu latarka / mama kazała mu przestać się wygłupiać i zgasić światło, od sytuacji w której kolega wpadł na pomysł, aby wysłać do ciebie bardzo długi ciąg zer? Być może musicie spróbować użyć jakiegoś kodowania.

### 4.3 Filozofowie (10 pkt)

$n$  filozofów siedzi przy bardzo dużym, okrągłym stole. Stół jest na tyle spory, że filozofowie mogą porozumiewać się tylko z pobliskimi filozofami.

Filozof jest zawsze głodny, jednak czasami jest to głód wiedzy, czasami zaś potrzeba ciała. Filozof może znajdować się więc w jednym z dwóch stanów - FILOZOFOWANIE lub JEDZENIE. Aby przejść do stanu JEDZENIE, filozof potrzebuje sztucców. Pomiedzy każdą parą filozofów znajduje się widelec. Filozof potrzebuje dwóch widelców do jedzenia. Jeśli sąsiedni filozof spożywa, filozof nie jest w stanie jeść (brakuje mu sztucca).

Filozofowie budzą się (to znaczy przestają rozważać sprawy wyższe) losowo i chcą przystąpić do jedzenia. Jeśli zaczną już jeść nie są chętni przestać.

Należy zaprojektować protokół o następujących właściwościach:

1. Jeśli w danym momencie są jacyś głodni filozofowie, któryś z nich JE.
2. Protokół na dłuższą metę nie faworyzuje żadnego z filozofów.
3. Protokół ma nie dopuszczać zakleszczeń (patrz punkt 1.)

Pamiętaj, że każdy filozof działa osobno, wiele rzeczy może więc się dziać równocześnie. Poniżej przykład niepoprawnego protokołu.

1. Jeśli prawy sztuciec wolny - weź. Wp.p. czekaj.
2. Jeśli lewy sztuciec wolny - weź i jedz. Wpp czekaj.

Jeśli wszyscy filozofowie obudzą się na raz, wszystkie sztucce będą podniesione. Filozofowie będą więc czekać w nieskończoność i żaden z nich nie zje.

## 5 Sieć

Do tego dorzucamy kilka zadań w celu ustalenia wspólnego języka. Odpowiedzi warto ułożyć własnymi słowami, bez przeklepania wikipedii. Warto skupić się na różnicach.

### 5.1 Urządzenia z lampkami (5 pkt)

Czym są modem, router i switch?

### 5.2 Adresacja (5 pkt)

Czym jest adres MAC i adres IP? Dlaczego nie wszystkie urządzenia w sieci mają adres IP?