Matematyka dyskretna

- Reprezentacja liczb w komputerze, liczby całkowite bez znaku i ze znakiem, układ uzupełnieniowy U2.
- Zasada szufladkowa Dirchleta. Zasada włączania-wyłączania.
- Rekursja, definicje rekurencyjne funkcji, rekurencja liniowa.
- Rekursywne typy danych, stosy i kolejki, algorytmy zdefiniowane rekurencyjnie, zasada dziel i rządź, sortowanie przez scalanie.
- Grafy nieskierowane i skierowane, drzewa. Drzewa binarnych przeszukiwań, algorytmy wstawiania i wyszukiwania. Notacja polska beznawiasowa.
- Spójność i acykliczność, drzewa rozpinające. Cykle i ścieżki Eulera i Hamiltona.
- Reprezentacja drzew i grafów. Algorytmy grafowe, przeszukiwanie w głąb i wszerz, konstruowanie drzew rozpinających, w tym minimalnych, szukanie najkrótszej ścieżki.

Sieci komputerowe

- Model ISO-OSI stosu protokołów komunikacyjnych
- Właściwości protokołu IP wersji 4 i 6
- Routing statyczny i dynamiczny
- Porównanie działania protokołów TCP i UDP
- Charakterystyka modelu klient-serwer
- Organizacja domen DNS w sieci Internet
- Protokół HTTP - bezstanowość, metody, najważniejsze nagłówki, metody uwierzytelniania, HTTPS

Wstęp do technologii web

- Responsywność strony - jakie mamy możliwości?
- Optymalizacja czasu ładowania strony
- Na czym polega kaskadowość stylów i ich właściwości?
- Pozycjonowanie - omów różnice pomiędzy wartościami static, relative, absolute i fixed
- Różnica między wartościami grid i flex właściwości display, przykłady użycia. Czy są sytuacje, gdzie możemy użyć jednej z nich, a drugiej nie (i odwrotnie)?
- CSS vs SCSS
- Możliwości języka SCSS na uzyskanie clean code

Bazy danych I

- Model relacyjnych baz danych. Własności relacji.
- Negatywne strony istnienia redundancji w bazie danych. Jakie są sposoby jej zwalczania? Przykłady.
- Rodzaje związków w teorii relacyjnych baz danych. w jaki sposób są one realizowane w języku SQL?
- Grupowanie danych w SQL. Zasada działania GROUP BY i HAVING.
- Pojęcie widoku (perspektywa, ang. view); zastosowania widoków.
- Sposoby łączenie tabel.

Matematyczne podstawy informatyki

- Omówić wyrażenia regularne i związek z deterministycznymi automatami skończonymi.
- Zastosowania wyrażeń regularnych.
- Własności języków regularnych.
- Omówić klasę języków bezkontekstowych oraz związek gramatyk bezkontekstowych i skończonych automatów ze stosem.
- Omówić maszyny Turinga i klasę języków akceptowanych przez maszyny Turinga.

Algorytmy i struktury danych

- Kopce binarne: definicja, ilustracja przykładem, operacja wstawiania i operacja usuwania elementu maksymalnego – idea i złożoność czasowa.
- Drzewa poszukiwań binarnych: definicja, ilustracja przykładem, operacje wstaw, szukaj, usuń – idea i złożoność czasowa.
- Algorytmy sortowania Quick-sort i sortowanie kopcowe (Heap-sort): idea algorytmów i omówić ich złożoność czasową
- Stosy, kolejki i kolejki priorytetowe: wyjaśnić typowe operacje wykonywane na tych strukturach danych (czyli Push i Pop dla stosów, wstawienie i pobranie elementu z kolejki dla kolejek, wstawienie i usunięcie elementu maksymalnego dla kolejek priorytetowych), podać przykładową implementację.
- Tablice z haszowaniem: idea haszowania i sposoby rozwiązywania kolizji (czyli metoda łańcuchowa i adresowanie otwarte).

Języki programowania I

- Czy da się programować bez wykorzystania mechanizmu „zmiennych”? Odpowiedź uzasadnij.
- Na czym polega rekurencja „ogonowa”? Podaj przykłady jej użycia.
- Omów podobieństwa i różnice pomiędzy strukturami listowymi (List[T]) i tablicowymi (Array[T]).
- Czym są funkcje „wyższego rzędu” i do czego mogą być przydatne? Odpowiedź zilustruj przykładami z użyciem języka Scala.
- Wymień i scharakteryzuj znane Ci rodzaje kolekcji w języku Scala. Odpowiedź zilustruj przykładami.
- Omów podstawowe pojęcia związane z programowaniem obiektowym w języku Scala: obiekt, klasa, cecha.
- Zilustruj je przykładami.
- Na czy polega model „aktorski” programowania równoległego? Czy aktor może wykonywać kilka akcji „jednocześnie”?

Języki programowania II

- Wskaż cechy programowania funkcyjnego i omów je na przykładzie języka JavaScript.
- Podaj przykłady funkcji wyższego rzędu (tj. funkcji, których argumentami są inne funkcje) na przykładzie 3 metod klasy Array: reduce, map, filter. Jaka jest semantyka tych 3 metod?
- Wskaż cechy programowania obiektowego na przykładzie języka JavaScript (notacja, tworzenie obiektów, dziedziczenie)
- Omów z przykładami typ Promise do asynchronicznych operacji w języku JavaScript

- Wyjaśnij z podaniem przykładów notację "arrow expression/fat arrow/lambda" w języku JavaScript. Jaką notację ta notacja zastępuje? Jakie ma ograniczenia (tj. gdzie nie powinno się stosować tej notacji)?

Systemy operacyjne

- Podstawowe i rozszerzone prawa dostępu w systemie Linux
- Procesy i wątki w systemie operacyjnym, techniki synchronizacji procesów i wątków, komunikacja międzyprocesowa
- Szeregowanie procesów: program szeregujący, algorytmy szeregowania
- Zakleszczenie zasobów: warunki powstania, techniki i algorytmy usuwania i unikania
- Wirtualizacja, hipernadzorca i podstawowe typy hipernadzorcy, techniki wirtualizacji: pełna wirtualizacja, parawirtualizacja (paravirtualization)
- Techniki ataków na systemy operacyjne (rodzaje złośliwego oprogramowania i inne popularne techniki), sposoby zabezpieczenia systemów komputerowych przed atakami

Protokoły sieci WEB

- Omów wykorzystanie kryptografii z kluczem publicznym w protokole SSH. Jakie znasz zastosowania kluczy SSH?
- Omów cechy i zastosowania protokołu TLS
- Omów protokół HTTP (zasada działania, zastosowania, kształt/budowa wymienianych danych)
- Omów wzorzec projektowy REST/RESTful HTTP API

Testowanie automatyczne

- Piramida testów i rola poszczególnych jej komponentów w zapewnianiu jakości oprogramowania
- Pojęcie długu technologicznego: konsekwencje i sposoby zapobiegania
- Test Driven Development w procesie rozwoju oprogramowania
- Wyjaśnij pojęcia: fake, stub, mock i spy, w kontekscie testowania oprogramowania
- Koncepcje Behavioral Driven Development na przykładzie biblioteki behave języka Python
- Tworzenie testów behaviouralnych w jezyku Gherkin

Bezpieczeństwo aplikacji webowych

- Wyjasnij pojęcia "uwierzytelnianie" i "autoryzacja" wg standardu OAuth2/OpenID Connect. Jakie znasz podstawowe terminy/pojęcia wykorzstywane przy specyfikacji flow/grant wg OAuth2?
- Wyjaśnij zastosowanie serwerów typu Indentity and Access Management (np. Keycloak, usługi Okta/Auth0). Jakie kroki/czynności należy wykonać żeby zastosować protokół OAuth2 i serwer/usługę IAM do zabezpieczenia API?
- Wyjaśnij jak działa "Authorization Code Flow/Grant" wg standardu OAuth2.

- Wyjaśnij jakie flow/grant wg standardu OAuth2 zastosujesz do komunikacji typu backend-to-backend/machine-to-matchine.
- Wyjaśnij zastosowanie techniki PKCE w "Authorization Code Flow/Grant" wg standardu OAuth2

Zarządzanie projektem informatycznym

- Zdefiniuj zarządzanie projektem, programem, portfelem
- Omów klasyfikacje projektów informatycznych
- Wymień charakterystyczne cechy projektów informatycznych
- Czynniki sukcesu i przyczyny porażek projektów informatycznych
- Wymień kluczowe obszary zarządzania projektem
- Wymień procesy zarządzania zakresem projektu i różnice pomiędzy zakresem projektu a zakresem produktu
- Dokonaj klasyfikacji otoczenia projektu IT i podaj przykłady interesariuszy
- Zdefiniuj ryzyko w projekcie i wymień techniki zarządzania ryzykiem

Inteligencja obliczeniowa

- Uczenie nadzorowane i nienadzorowane - definicje, przykłady
- Dobór cech a odkrywanie cech w data science
- Czego dotyczy i na czym polega algorytm propagacji wstecznej?
- Regresja a regresja logistyczna.
- Klasteryzacja hierarchiczna - do czego służy, rodzaje, korzyści ze stosowania
- Krzywe ROC - co to są i do czego służą?

Programowanie obiektowo-funkcyjne

- Pojęcie klasy i obiektu w językach programowania na wybranych przykładach
- Wyjaśnij pojęcie polimorfizmu na przykładach kodu w języku Java
- Pojęcie interfejsu w języku Java, jego implementacja i znaczenie w zasadzie odwracania zależności (dependency inversion principle)
- Lambda-wyrażenia w języku Java i ich zastosowanie w przetwarzaniu strumieni
- Znaczenie zasady pojedynczej odpowiedzialności (single responsibility principle) i otwarte-zamknięte (open closed principle) w tworzeniu testowalnego kodu obiektowego
- Na wybranych przykładach wyjaśnij zasadę segregacji interfejsów (interface segregation principle) w programowaniu obiektowym

Analiza i projektowanie systemów informatycznych

- Istota i sposoby dokumentowania wymagań systemowch
- Sposoby odwzorowywania podstawowych pojęć obiektowości na diagramach klas UML
- Diagramy czynności a diagramy sekwencji UML - zastosowanie, podobieństwa i różnice
- Narzędzia CASE - zastosowanie i typowa funkcjonalność
- Główne fazy procesu wytwórczego oprogramowania