

KONSPEKT PRZEDMIOTU

Semestr letni, rok akad. 2020/2021

Nazwa przedmiotu	Algorytmy i złożoność
Kierunek/-i studiów / rok studiów / semestr studiów	Informatyka/rok 1/semestr 2
Tryb studiów	Stacjonarny
Forma zajęć	Wykład
Liczba godzin	30
Koordinator przedmiotu:	dr Feliks Kurp
Jakie są ogólne cele dydaktyczne tego przedmiotu?	<ol style="list-style-type: none"> 1. Nabycie umiejętności rozwiązywanie problemów poprzez poprawną algorytmizację. 2. Nabycie teoretycznej i praktycznej umiejętności posługiwania się złożonymi strukturami danych. 3. Nabycie umiejętności oceny klasy algorytmów z podstawami szacowania ich złożoności obliczeniowej.
Jak są kryteria zaliczenia tego przedmiotu?	<p>Aby zaliczyć wykład należy zdać egzamin. Przewiduje się egzamin zerowy na ostatnim wykładzie. Przystąpienie do egzaminu zerowego nie jest obowiązkowe. Po zakończeniu semestru odbędą się egzaminy podstawowy i poprawkowy w terminach określonych w harmonogramie sesji letniej.</p> <p>Egzamin polega na odpowiedzi na 6 pytań otwartych w ograniczonym czasie. Celem egzaminu jest sprawdzenie zrozumienia zagadnień omawianych na wykładzie, a także (w pewnym zakresie) sprawdzenie umiejętności praktycznych. Maksymalna ocena zaliczająca wykład to ocena <i>bardzo dobry</i> (5.0).</p>
Jakie są kryteria zaliczenia tego przedmiotu na ocenę celującą?	<p>Metodą zwiększania oceny z zaliczenia wykładu do oceny celującej (6.0) jest aktywność studentów w trakcie wykładu. Otrzymają ją studenci (którzy zaliczyli egzamin podstawowy na 5.0) za interesujące wypowiedzi, lub też zadawanie ciekawych pytań wykładowcy w trakcie wykładów.</p>
Pozostałe informacje, dotyczące tego przedmiotu, ważne dla studenta	<p>Przedmiot pozwala wyrobić w sobie umiejętność rozwiązywania złożonych problemów poprzez poprawnie przeprowadzoną algorytmizację drogą logicznego, konsekwentnego rozumowania.</p> <p>Przedmiot pozwala też poznać własności podstawowych, używanych w programowaniu, struktur danych. Są to umiejętności niezbędne w każdej dziedzinie informatyki.</p> <p>Ważną umiejętnością jest też ocena klasy algorytmów poprzez szacowanie ich złożoności obliczeniowej.</p>

ZAJĘCIA 1 (3 godz.)

Temat: **Pojęcie i własności algorytmów. Schemat konstruowania poprawnych algorytmów**

Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał)	<p>Efekty uczenia się:</p> <p>Po tych zajęciach student</p> <ol style="list-style-type: none"> 1. będzie rozumiał na czym polega przetwarzanie imperatywne, 2. będzie wiedział na czym polega konieczność określenia asercji początkowej i końcowej w algorytmie, 3. będzie znał podstawowe metody zapisu i analizy poprawności algorytmu, 4. będzie umiał zapisać algorytm z wykorzystaniem pseudokodu.
Treści zajęć	<ol style="list-style-type: none"> 1. Przetwarzanie imperatywne. 2. Asercja początkowa i końcowa algorytmu. 3. Postacie algorytmu. 4. Pseudokod. 5. Typowe przykłady algorytmów.
Źródło/a do nauki treści z tych zajęć	<p>Obowiązkowe:</p> <p>- AiZ wykład część 1, rozdział 1 i 4</p> <p>Uzupełniające:</p> <p>- Piotr Wróblewski: Algorytmy struktury danych i techniki programowania, Wydanie V Helion</p>

ZAJĘCIA 2 (3 godz.)

Tematy: **Algorytmy iteracyjne**

Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał)	<p>Efekty uczenia się:</p> <p>Po tych zajęciach student</p> <ol style="list-style-type: none"> 1. będzie umiał budować algorytmy z rozgałęzieniami oparte na operacjach warunkowych, 2. będzie wiedział jaką wybrać pętlę iteracyjną w zależności charakteru rozwiązywanego problemu, 3. będzie wiedział jak wykażać zachodzenie warunku stopu w algorytmach iteracyjnych, 4. będzie umiał szacować złożoność obliczeniową prostych algorytmów iteracyjnych.
Treści zajęć	<ol style="list-style-type: none"> 1. Instrukcja warunkowa if. 2. Trójargumentowa operacja warunkowa. 3. Pętle iteracyjne, warunki stopu pętli iteracyjnej. 4. Podstawy szacowania złożoności obliczeniowej algorytmu.
Źródło/a do nauki treści z tych zajęć	<p>Obowiązkowe:</p> <p>- AiZ wykład część 1 rozdział 2</p>

ZAJĘCIA 3 (3 godz.) Tematy: Algorytmy rekurencyjne	
Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał)	<p>Efekty uczenia się: Po tych zajęciach student</p> <ol style="list-style-type: none"> 1. będzie umiał zapisać proces rekurencyjny w postaci funkcji rekurencyjnej, 2. będzie wiedział jaką rolę odgrywają parametry funkcji rekurencyjnej w przebiegu procesu rekurencyjnego, 3. będzie rozumiał jakie zjawiska zachodzą w trakcie realizacji procesu rekurencyjnego, 4. będzie wiedział jakie są zasadnicze różnice między algorytmami rekurencyjnymi ogonowymi i bezogonowymi, 5. będzie potrafił przekształcić prosty algorytm iteracyjny w rekurencyjny i odwrotnie
Treści zajęć	<ol style="list-style-type: none"> 1. Funkcje rekurencyjne. 2. Komunikacja funkcji rekurencyjnej z otoczeniem, efekty uboczne, 3. Rekurencja a iteracja, 4. Algorytmy rekurencyjne ogonowe i bezogonowe, 5. Przykłady algorytmów rekurencyjnych.
Źródło/a do nauki treści z tych zajęć	<p>Obowiązkowe: - AiZ wykład część 1 rozdział 3</p>

ZAJĘCIA 4 (3 godz.) Tematy: Typy proste i złożone. Tablice indeksowane. Tablice asocjacyjne.	
Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał)	<p>Efekty uczenia się: Po tych zajęciach student</p> <ol style="list-style-type: none"> 1. będzie rozumiał na czym polega różnica między prostym typem wartościowym i typem referencyjnym, 2. będzie wiedział jak utworzyć i obsługiwać algorytmicznie jednowymiarową tablicę indeksowaną, 3. będzie rozumiał jak się tworzy, na czym polega i jak wykorzystać w programowaniu dostęp bezpośredni w tablicy indeksowanej, 4. będzie wiedział jak utworzyć i wykorzystać w programowaniu tablicę dwuwymiarową, 5. będzie rozumiał jakie są różnice między tablicą statyczną a dynamiczną. 6. będzie wiedział, jaka jest budowa, własności i zastosowania tablicy asocjacyjnej.
Treści zajęć	<ol style="list-style-type: none"> 1. Proste typy wartościowe i referencyjne, 2. Typ złożony – jednowymiarowa tablica indeksowana, 3. Tworzenie i własności tablicy indeksowanej, 4. Dostęp bezpośredni, 5. Tablice wielowymiarowe i dynamiczne, 6. Tablice asocjacyjne (słowniki).
Źródło/a do nauki treści z tych zajęć	<p>Obowiązkowe: - AiZ wykład część 2, rozdziały 5.1 i 5.2</p>

ZAJĘCIA 5 (3 godz.) Tematy: Podstawowe klasy algorytmów i ich cechy.	
Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał)	Efekty uczenia się: Po tych zajęciach student: <ol style="list-style-type: none"> 1. będzie rozumiał jaka jest różnica w złożoności obliczeniowej algorytmów różnych klas i od czego ona zależy. 2. będzie wiedział na czym polega sortowanie indeksowe, 3. będzie rozumiał, na przykładzie algorytmu <i>QuickSort</i>, na czym polega idea algorytmów typu „dziel i zwyciężaj”,
Treści zajęć	<ol style="list-style-type: none"> 1. Wyszukiwanie liniowe i binarne. 2. Algorytmy sortowania tablic. Sortowanie indeksowe. 3. Metoda ”dziel i zwyciężaj”. Przykładowy schemat algorytmu <i>QuickSort</i>. 4. Złożoność obliczeniowa algorytmów różnych klas.
Źródło/a do nauki treści z tych zajęć	Obowiązkowe: - AiZ wykład część 2 rozdział 6

ZAJĘCIA 6 (3 godz.) Tematy: Procesy rekurencyjne.	
Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał)	Efekty uczenia się: Po tych zajęciach student: <ol style="list-style-type: none"> 1. będzie rozumiał znaczenie pojęć: głębokość rekurencji, liczba wywołań rekurencyjnych, maksymalna zajętość pamięci w rekurencji, 2. będzie rozumiał znaczenie rekurencji w obsłudze struktur dynamicznych, takich jak drzewa binarne, czy grafy, 3. będzie znał procesy zachodzące na stosie dla zmiennych w rekurencji i ich wykorzystanie w algorytmice, 4. będzie umiał oszacować złożoność obliczeniową dla różnych rodzajów procesów rekurencyjnych, 5. będzie wiedział, jak dokonać derekursywacji dla wybranych algorytmów rekurencyjnych.
Treści zajęć	<ol style="list-style-type: none"> 1. Zjawiska na stosie dla zmiennych w rekurencji. 2. Anatomia wywołania rekurencyjnego. 3. Rekurencja pośrednia i zagnieżdżona. 4. Implementacja stosowa rekurencji. 5. Derekursywacja.
Źródło/a do nauki treści z tych zajęć	Obowiązkowe: - AiZ wykład część 2 rozdział 7

ZAJĘCIA 7 (3 godz.)

Tematy: Programowanie z wykorzystaniem struktur liniowych opartych na referencji.

Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał)	<p>Efekty uczenia się:</p> <p>Po tych zajęciach student:</p> <ol style="list-style-type: none"> 1. będzie znał i rozumiał pojęcia: typ referencyjny, zmienna referencyjna, rekurencyjna definicja obiektu z referencją. 2. będzie znał własności i podstawowe algorytmy obsługi list liniowych jedno i dwukierunkowych. 3. będzie potrafił zapisać algorytm wykorzystujący dynamiczny LIFO-stos, lub FIFO-kolejkę. 4. będzie znał własności samoorganizujących się list. 5. będzie znał metodę konstrukcji list z przeskokami.
Treści zajęć	<ol style="list-style-type: none"> 1. Typy i zmienne referencyjne. 2. Własności struktur dynamicznych. 3. Obsługa list liniowych jedno i dwukierunkowych. 4. Dynamiczne LIFO-stosy i FIFO-kolejki. 5. Samoorganizujące się listy. 6. Przeszukiwanie indeksowo-sekwencyjne (listy z przeskokami).
Źródło/a do nauki treści z tych zajęć	<p>Minimalne:</p> <p>- AiZ wykład część 3</p>

ZAJĘCIA 8 (3 godz.)

Tematy: Drzewa i lasy. Drzewo binarne.

Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał)	<p>Efekty uczenia się:</p> <p>Po tych zajęciach student:</p> <ol style="list-style-type: none"> 1. będzie znał i rozumiał rekurencyjną definicję dowolnego drzewa. 2. będzie umiał posługiwać się pojęciami dotyczącymi drzew, 3. będzie znał i rozumiał działanie rekurencyjnych algorytmów przeszukiwania drzew binarnych: <i>preorder</i>, <i>inorder</i> i <i>postorder</i>, 4. będzie znał i rozumiał konstrukcję drzewa <i>BST</i> i jego znaczenie w sortowaniu danych, 5. będzie znał i rozumiał znaczenie procesu równoważenia drzew.
Treści zajęć	<ol style="list-style-type: none"> 1. Rekurencyjna definicja drzewa. 2. Definicje podstawowych pojęć dotyczących drzew i lasów. 3. Drzewa binarne - własności. 4. Algorytmy przeszukiwania drzew binarnych. 5. Drzewo binarnych przeszukiwań (<i>BST</i>) – definicja i własności. 6. Złożoność obliczeniowa algorytmu przeszukiwania drzewa <i>BST</i>. 7. Drzewa wyważone i dokładnie wyważone (<i>AVL</i>). Równoważenie drzew.
Źródło/a do nauki treści z tych zajęć	<p>Obowiązkowe:</p> <p>- AiZ wykład część 4</p>

ZAJĘCIA 9 (3 godz.) Tematy: Drzewa z priorytetem (HPO-drzewa – kopce). Grafy.	
Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał)	Efekty uczenia się: Po tych zajęciach student: <ol style="list-style-type: none"> 1. będzie znał własności drzewa <i>HPO</i> i jego znaczenie, 2. będzie potrafił wskazać miejsce wstawienia nowego wężła do drzewa <i>HPO</i> z uwzględnieniem stopnia wyważenia drzewa. 3. będzie potrafił ocenić własności grafu skierowanego, takie jak: cykliczność, płaskość, spójność i silna spójność grafu, 4. będzie wiedział jakie są metody reprezentacji grafu w pamięci operacyjnej, oraz ich własności, 5. będzie rozumiał jak jest zbudowana reprezentacja grafu w postaci listy incydencji, 6. będzie rozumiał budowę i działanie rekurencyjnego algorytmu szukania w głąb dla grafu, oraz jego znaczenie praktyczne.
Treści zajęć	<ol style="list-style-type: none"> 1. Drzewo z priorytetem – własności. 2. Znaczenie drzewa z priorytetem w praktyce informatycznej. 3. Grafy – podstawowe pojęcia. 4. Grafy – metody reprezentacji w pamięci. 5. Dynamiczna reprezentacja grafu w postaci listy incydencji. 6. Algorytm szukania w głąb dla grafu jako przykład algorytmu z powrotami.
Źródło/a do nauki treści z tych zajęć	Obowiązkowe: - AiZ wykład część 5

ZAJĘCIA 10 (3 godz.) Tematy: Algorytmy z powrotami. Problemy algorytmicznie trudne. Egzamin zerowy.	
Po tych zajęciach student (będzie znał/ wiedział/ umiał/ potrafił / rozumiał)	Efekty uczenia się: Po tych zajęciach student: <ol style="list-style-type: none"> 1. będzie potrafił zidentyfikować klasę algorytmów z powrotami i wymienić najbardziej znane algorytmy z tej klasy, 2. będzie rozumiał ideę ogólnej postaci algorytmu z powrotami, 3. będzie wiedział jak zaimplementować algorytm z powrotami do rozwiązania problemów wyszukiwania wyczerpującego. 4. będzie wiedział jakie algorytmy zaliczane są w informatyce teoretycznej do problemów algorytmicznie trudnych, co to są problemy <i>NP</i>, <i>NP.-zupelne</i> i <i>NP.-trudne</i>.
Treści zajęć	<ol style="list-style-type: none"> 1. Klasa algorytmów z powrotami i przykłady ich użycia. 2. Ogólna postać rekurencyjnego algorytmu z powrotami. 3. Implementacje rekurencyjnego algorytmu z powrotami. 4. Problemy algorytmicznie trudne. 5. Egzamin zerowy.
Źródło/a do nauki treści z tych zajęć	Obowiązkowe: - AiZ wykład część 5