



Algorytmy i struktury danych

Karta opisu przedmiotu

Informacje podstawowe

Kierunek studiów Informatyka (kierunek wspólny - WI)	Cykl dydaktyczny 2020/2021	
Specjalność -	Kod przedmiotu WIINF5.II2.00477.20	
Jednostka organizacyjna Wydział Informatyki	Języki wykładowe polski	
Poziom kształcenia Studia inżynierskie I stopnia	Obligatoryjność Obowiązkowy	
Forma studiów Stacjonarne	Blok zajęciowy Przedmioty ogólne	
Profil studiów Ogólnoakademicki	Przedmiot powiązany z badaniami naukowymi Tak	
Koordynator przedmiotu	Piotr Faliszewski	
Prowadzący zajęcia	Piotr Faliszewski, Marcin Kurdziel, Marek Gajęcki	
Okres Semestr 2	Forma zaliczenia Egzamin	Liczba punktów ECTS 6
	Forma prowadzenia i godziny zajęć Wykład: 30 Ćwiczenia audytoryjne: 30	

Efekty uczenia się dla przedmiotu

Kod	Efekty w zakresie	Kierunkowe efekty uczenia się	Metody weryfikacji
Wiedzy - Student zna i rozumie:			
W1	Zna i rozumie podstawowe mechanizmy i techniki projektowania algorytmów.	INF1A_W02	Egzamin

Kod	Efekty w zakresie	Kierunkowe efekty uczenia się	Metody weryfikacji
W2	Zna i rozumie podstawowe techniki analizy algorytmów.	INF1A_W02	Egzamin
W3	Ma wiedzę dotyczącą standardowych struktur danych.	INF1A_W02	Egzamin
W4	Ma wiedzę dotyczącą podstawowych algorytmów.	INF1A_W02	Egzamin
Umiejętności - Student potrafi:			
U1	Potrafi rozwiązywać problemy algorytmiczne.	INF1A_U05	Kolokwium
U2	Potrafi stosować standardowe struktury danych.	INF1A_U05	Kolokwium
Kompetencje społecznych - Student jest gotów do:			
K1	Potrafi krytycznie ocenić przedstawione rozwiązania algorytmiczne, uzasadnić swoją opinię i rozszerzyć rozwiązania sugerowane w dyskusji grupowej.	INF1A_K03	Udział w dyskusji

Treści programowe zapewniające uzyskanie efektów uczenia się dla modułu zajęć

Przedmiot zapoznaje studentów z podstawowymi algorytmami, strukturami danych, oraz metodami tworzenia i analizy algorytmów.

Nakład pracy studenta

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane aktywności
Wykład	30
Ćwiczenia audytoryjne	30
Przygotowanie do zajęć	30
Samodzielne studiowanie tematyki zajęć	55
Dodatkowe godziny kontaktowe	5
Łączny nakład pracy studenta	Liczba godzin 150
Liczba godzin kontaktowych	Liczba godzin 60

* godzina (lekcyjna) oznacza 45 minut

Treści programowe

Lp.	Treści programowe	Efekty uczenia się dla przedmiotu	Formy prowadzenia zajęć
1.	<p>1. Wprowadzenie do algorytmiki (1 godz.) Analiza algorytmu, złożoność obliczeniowa, notacja asymptotyczna, praktyczna i teoretyczna analiza złożoności obliczeniowej algorytmu.</p> <p>2. Algorytmy sortowania o złożoności $O(n \log n)$ (3 godz.) Algorytmy MergeSort, HeapSort i kopiec binarny, QuickSort, statystyki pozycyjne</p> <p>3. Algorytmy sortowania o złożoności liniowej (1 godz.) Sortowanie przez zliczanie, sortowanie kubełkowe, sortowanie pozycyjne</p> <p>4. Elementarne struktury danych (4 godz.) Tablica, lista, drzewo (w tym BST, czerwono-czarne), tablica z haszowaniem, B-drzewa (1 godz.). Abstrakcyjne struktury danych i ich implementacja (stos, kolejka, tablica asocjacyjna)</p> <p>5. Drzewa przedziałowe (2 godz.) Rodzaje drzew przedziałowych, implementacja, zastosowania.</p> <p>6. Reprezentacja grafów i elementarne algorytmy grafowe (3 godz.) Reprezentacja grafów. Algorytmy BFS, DFS, zastosowania (najkrótsze ścieżki bez wag, sortowanie topologiczne, silnie spójne składowe, mosty)</p> <p>7. Najkrótsze ścieżki oraz minimalne drzewo rozpinające (5 godz.) Algorytmy Dijkstry, Bellmana-Forda, Kruskala, Prima (w tym lasy zbiorów rozłącznych), Floyda-Warshalla,</p> <p>8. Programowanie dynamiczne (2 godz.) Zasada programowania dynamicznego, ciąg Fibonacciego, najdłuższy rosnący podciąg, problem plecakowy, zbiór wierzchołków niezależnych w drzewie</p> <p>9. Algorytmy zachłanne (2 godz.) Zasada działania algorytmów zachłannych, przykłady (kody Huffmana, problem wyboru zajęć i inne). Dowodzenie poprawności algorytmów zachłannych</p> <p>10. Zaawansowane techniki algorytmiczne (3 godz.) Zaawansowane przykłady stosowania metod algorytmicznych. Algorytmy randomizowane, algorytmy aproksymacyjne.</p> <p>11. Wprowadzenie do algorytmów geometrycznych (2 godz.) Najbliższe dwa punkty, para przecinających się odcinków, otoczka wypukła</p> <p>12. Wprowadzenie do algorytmów tekstowych (2 godz.) Problem wyszukiwania wzorca, algorytm KMP</p>	W1, W2, W3, W4	Wykład

Lp.	Treści programowe	Efekty uczenia się dla przedmiotu	Formy prowadzenia zajęć
2.	1. Proste metody sortowania (2 godz.) 2. Sortowania szybkie i liniowe cz. 1 (2 godz.) 3. Sortowania szybkie i liniowe cz. 2 (2 godz.) 4. Elementarne struktury danych (2 godz.) 5. Kolokwium 1 (2 godz.) 6. Wykorzystanie struktur danych (2 godz.) 7. Drzewa przedziałowe (2 godz.) 8. Algorytmy grafowe (DFS, BFS) (2 godz.) 9. Algorytmy na grafach ważonych (2 godz.) 10. Kolokwium 2 (2 godz.) 11. Wykorzystanie algorytmów grafowych (2 godz.) 12. Programowanie zachłanne i dynamiczne (2 godz.) 13. Wykorzystanie metod konstrukcji algorytmów (2 godz.) 14. Kolokwium 3 (2 godz.) 15. Algorytmy geometryczne (2 godz.)	U1, U2, K1	Ćwiczenia audytoryjne

Informacje rozszerzone

Metody i techniki kształcenia :

Mini wykład

Rodzaj zajęć	Metody zaliczenia	Warunki zaliczenia przedmiotu
Wykład	Udział w dyskusji, Egzamin	
Ćwiczenia audytoryjne	Udział w dyskusji, Kolokwium	

Warunki i sposób zaliczenia poszczególnych form zajęć, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu

Warunku zaliczenia ćwiczeń: W trakcie ćwiczeń odbywają się trzy kolokwia. Zaliczenie ćwiczeń wymaga uzyskania sumy punktów z tych ćwiczeń (oraz dodatkowych punktów uzyskanych w trakcie odpowiedzi ustnych) powyżej ustalonego progu (ale nie większego niż połowa możliwych do uzyskania punktów). Zaliczenie egzaminu: Uzyskanie liczby punktów na egzaminie pisemnym powyżej ustalonego progu (nie większego niż połowa maksymalnej możliwej liczby punktów). Zaliczenia poprawkowe mają postać pisemnych kolokwiów zaliczeniowych. Student dopuszczany jest do egzaminu dopiero po zaliczeniu ćwiczeń (tracąc terminy egzaminu, które odbyły się przed zaliczonym przez niego kolokwium zaliczeniowym).

Sposób obliczania oceny końcowej

Ocena końcowa jest średnią wszystkich ocen (wliczając wszystkie terminy) otrzymane z ćwiczeń i egzaminów (egzaminu nie pisane ze względu na brak zaliczenia ćwiczeń otrzymują ocenę 2.0). Jeśli średnia nie jest regulaminową oceną, to ocena zaokrąglana jest w stronę korzystniejszą dla studenta.

Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach

Wyrównywanie zaległości powstałych przez nieobecność studenta na zajęciach: Jeśli student opuści dane zajęcia to może brać udział w zajęciach dla innej grupy. Jeśli jest to niemożliwe, student jest zobowiązany do samodzielnego uzupełnienia wiedzy. W przypadku nieobecności na kolokwiach, student jest zobowiązany do zaliczenia odpowiedniego materiału w trakcie kolokwium zaliczeniowego.

Wymagania wstępne i dodatkowe

Znajomość matematyki (systemy pozycyjne, kombinatoryka, logarytmy, itp.)
 Podstawowa znajomość programowania w C.

Zasady udziału w poszczególnych zajęciach, ze wskazaniem, czy obecność studenta na zajęciach jest obowiązkowa

Wykład: Studenci uczestniczą w zajęciach poznając kolejne treści nauczania zgodnie z sylabusem przedmiotu. Studenci winni na bieżąco zadawać pytania i wyjaśniać wątpliwości. Rejestracja audiowizualna wykładu wymaga zgody prowadzącego. Ćwiczenia audytoryjne: Studenci przystępując do ćwiczeń są zobowiązani do przygotowania się w zakresie wskazanym każdorazowo przez prowadzącego (np. w formie zestawów zadań). Ocena pracy studenta może bazować na wypowiedziach ustnych lub pisemnych w formie kolokwium, co zgodnie z regulaminem studiów AGH przekłada się na ocenę końcową z tej formy zajęć.

Literatura

Obowiązkowa

1. References:
2. 1. „Wprowadzenie do algorytmów” Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Wydawnictwo Naukowe PWN, 2012. (Polish translation of the “Introduction to Algorithms” textbook)
3. Additional References:
4. 2. „Algorytmy” R. Sedgewick, K. Wayne, Helion 2012.
5. 3. „Algorytmy” S. Dasgupta, C. Papadimitriou, U. Vazirani, Wydawnictwa Naukowe PWN 2010.

Badania i publikacje

Publikacje

1. 1. P. Faliszewski, R. Niedermeier, Parameterization in Computational Social Choice, In M-Y. Kao, editor, Encyclopedia of Algorithms, Springer, 2015.
2. 2. E. Elkind, P. Faliszewski: Approximation Algorithms for Campaign Management. Proceedings of the 6th International Workshop on Internet and Network Economics, 2010, pp. 473-482
3. 3. P. Skowron, L. Yu, P. Faliszewski, E. Elkind: The complexity of fully proportional representation for single-crossing electorates. Theor. Comput. Sci. 569: 43-57 (2015)

Kierunkowe efekty uczenia się

Kod	Treść
INF1A_K03	Potrafi współdziałać i pracować w grupie, przyjmując w niej różne role, ma świadomość odpowiedzialności za pracę własną i za wspólnie realizowane zadania.
INF1A_U05	Potrafi wykorzystać poznane metody i modele do tworzenia programów o charakterze użytkowym, a także potrafi adekwatnie wykorzystać znane algorytmy i struktury danych w budowie systemu komputerowego.
INF1A_W02	Ma szczegółową wiedzę w zakresie podstaw algorytmiki, struktur danych oraz złożoności obliczeniowej, a także w zakresie podstaw teoretycznych budowy wybranych narzędzi i systemów informatycznych.