

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Praktyka i teoria szeregowania zadań		Kod 1010511371010510101
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 4 / 7
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obieralny
Stopień studiów: I stopień	Forma studiów (stacjonarna/niestacjonarna) stacjonarna	
Godziny Wykłady: 30 Ćwiczenia: - Laboratoria: 30 Projekty/seminaria: -		Liczba punktów 4
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) kierunkowy		(ogólnouczelniany, z innego kierunku) z danego kierunku
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne nauki techniczne		Podział ECTS (liczba i %) 4 100% 4 100%
Odpowiedzialny za przedmiot / wykładowca:		
dr hab. inż. Małgorzata Sterna, prof. nadzw. email: Małgorzata.Sterna@cs.put.poznan.pl tel. +48 61 6652982 Wydział Informatyki 60-965 Poznań, ul. Piotrowo 2		dr Maciej Machowiak email: Maciej.Machowiak@put.poznan.pl tel. +48 61 6652982 Wydział Informatyki 60-965 Poznań, ul. Piotrowo 2
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student rozpoczynający przedmiot powinien posiadać podstawową wiedzę z zakresu algorytmów i struktur danych, złożoności obliczeniowej oraz programowania w dowolnym języku wysokiego poziomu.
2	Umiejętności:	Winien posiadać umiejętność analizy zagadnień o charakterze praktycznym, rozwiązywania problemów kombinatorycznych, zaimplementowania algorytmów rozwiązujących takie problemy, oceny jakości algorytmów, a także umiejętność pozyskiwania informacji ze wskazanych źródeł.
3	Kompetencje społeczne	Ponadto student powinien wykazywać się samodzielnością, odpowiedzialność za pracę wykonywaną indywidualnie oraz w grupie, zdolnością prezentacji w formie ustnej i pisemnej ogólnej koncepcji i sposobu rozwiązania analizowanych problemów.
Cel przedmiotu:		
1. Przekazanie studentom podstawowej wiedzy na temat problematyki szeregowania zadań na przykładzie podstawowych modeli z pojedynczą maszyną, maszynami równoległymi i dedykowanymi.		
2. Przedstawienie wybranych algorytmów szeregowania zadań, wykorzystujących różne techniki algorytmiczne m.in. z zakresu teorii grafów czy geometrii oraz prezentacja sposobu analizy i rozwiązywania zagadnień szeregowania z wykorzystaniem osiągnięć teorii złożoności obliczeniowej.		
3. Rozwijanie u studentów umiejętności rozwiązywania problemów szeregowania od etapu modelowania zagadnienia praktycznego, przez dobór/projektowanie algorytmu, po ocenę jego efektywności w eksperymencie obliczeniowym.		
4. Kształtowanie u studentów umiejętności przeprowadzania prostych eksperymentów obliczeniowych i opracowywania ich wyników oraz umiejętności pracy zespołowej w trakcie rozwiązywania zadań podczas zajęć laboratoryjnych.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
1. Student ma wiedzę ogólną dotyczącą podstaw teorii szeregowania zadań ze szczególnym uwzględnieniem systemów obsługi czyli systemów z procesorami dedykowanymi - [K1st_W4]		
2. zna klasyczne algorytmy szeregowania zadań w szczególności dla pojedynczej maszyny, maszyn równoległych i dla dwumaszynowych systemów obsługi - [K1st_W5]		
3. zna przykłady wykorzystania różnych technik algorytmicznych oraz metod złożoności obliczeniowej w rozwiązywaniu problemów szeregowania zadań - [K1st_W7]		
Umiejętności:		

<ol style="list-style-type: none">1. Student potrafi zaproponować algorytmy, głównie heurystyczne, rozwiązujące problem szeregowania zadań - [K1st_U10]2. potrafi zaimplementować i ocenić efektywność zaproponowanego algorytmu z różnych punktów widzenia - [K1st_U11]3. potrafi uczestniczyć w dyskusji nad różnymi koncepcjami rozwiązania problemu szeregowania zadań i oceną ich jakości - [K1st_U2]4. potrafi zaprojektować i przeprowadzić eksperyment obliczeniowy mający na celu ocenę efektywności prostych algorytmów szeregowania zadań - [K1st_U4]5. potrafi dostosować dostępne w literaturze algorytmy i metody do rozwiązania postawionego problemu szeregowania - [K1st_U19]6. potrafi zaprezentować wybraną przez siebie koncepcję rozwiązania problemu ze wskazaniem jej słabych i mocnych stron oraz uczestniczyć w dyskusji nad rozwiązaniami innych osób - [K1st_U1]
Kompetencje społeczne:
<ol style="list-style-type: none">1. Student ma świadomość ciągłego rozwoju technik algorytmicznych wykorzystywanych do rozwiązywania problemów praktycznych - [K1st_K1]2. ma świadomość wsparcia jakie w rozwiązywaniu problemów praktycznych dostarczają wyniki badań naukowych - [K1st_K2]

Sposoby sprawdzenia efektów kształcenia
<p>Ocena formująca:</p> <ol style="list-style-type: none">a) ocenianie ciągle w trakcie wykładów, poprzez premiowanie aktywnego uczestnictwa w zajęciach, przejawiającego się udziałem w dyskusji inicjowanej przez wykładowcę, zgłaszaniem uwag umożliwiających doskonalenie materiałów dydaktycznych i sposobu prezentacji treści programowych;b) ocenianie ciągle w trakcie laboratoriów na podstawie oceny bieżącego postępu realizacji zadań, udziału w dyskusjach na temat proponowanych rozwiązań i sposobu przeprowadzania eksperymentów obliczeniowych, w tym premiowanie przyrostu umiejętności posługiwania się poznanymi technikami i algorytmami koniecznymi do realizacji zadań oraz biegłości w implementacji algorytmów. <p>Ocena podsumowująca:</p> <p>Sprawdzanie założonych efektów kształcenia realizowane jest przez:</p> <ol style="list-style-type: none">a) ocenę wiedzy i umiejętności wykazanych na kolokwium zaliczeniowym przeprowadzonym pod koniec semestru, mającym formę testu, ale wymagających samodzielnego zastosowania np. technik algorytmicznych i metod modelowania prezentowanych w trakcie wykładu;b) do uzyskania oceny dostatecznej z wykładu konieczne jest przekroczenie progu 50% łącznej liczby punktów, progi kolejnych ocen ulegają zwiększeniu o 10%;c) ocenę wykonania ćwiczeń laboratoryjnych, obejmującą ocenę sprawozdań, w tym ocenę sposobu prezentacji zaproponowanych metod oraz przeprowadzonych eksperymentów; umiejętności zwięzłego i jasnego prezentowania algorytmów, szacowania ich złożoności obliczeniowej, poprawnego wykonywania eksperymentów obliczeniowych oraz prezentacji ich wyników w formie pisemnej;d) ocenę sposobu prezentacji zaproponowanych rozwiązań i uzyskanych wyników w formie ustnej w trakcie zajęć;e) ocenę umiejętności pracy w zespole i uczestnictwa w dyskusji prowadzonej na forum grupy laboratoryjnej;f) ocenę dojrzałości i odpowiedzialności studentów przejawiającej się m.in. odpowiednią frekwencją na zajęciach oraz terminowością wykonywania zadań. <p>Dodatkowe premiowanie aktywności podczas zajęć, w tym:</p> <ol style="list-style-type: none">a) uczestnictwa w dyskusjach;b) przedstawienia interesujących pomysłów rozwiązania postawionych problemów;c) wykazania się ponadprogramowymi umiejętnościami, w tym z zakresu algorytmiki, syntezy poznanych technik i metod oraz analizy zagadnień będących tematem zajęć;d) wnikliwego śledzenia prezentowanych treści w stopniu umożliwiającym wykrycie ewentualnych nieścisłości czy też zgłaszania propozycji udoskonalenia materiałów dydaktycznych;e) wskazywania trudności percepcyjnych studentów, umożliwiające bieżące doskonalenia procesu dydaktycznego.
Treści programowe
<p>W ramach wykładu przedstawiane są następujące zagadnienia:</p> <ol style="list-style-type: none">1. Ogólna definicja problemu szeregowania zadań, podstawowe parametry zadań i procesorów oraz funkcje kryterialne wraz z ich interpretacją praktyczną.2. Klasyfikacja problemów szeregowania zadań z użyciem notacji trójpolowej wraz z jej odniesieniem do problemów spotykanych w praktyce.3. Przykłady dowodzenia trudności zagadnień szeregowania, klasyfikacja problemów szeregowania z uwagi na złożoność obliczeniową.4. Wybrane problemy szeregowania na pojedynczej maszynie, algorytmy listowe, algorytm Jacksona, algorytm Horna, reguła Smitha, algorytm Hodgsona.5. Wybrane problemy szeregowania na maszynach równoległych, algorytm McNaughtona, algorytm Hu, algorytmy listowe,

- przykłady algorytmów aproksymacyjnych, programowania dynamicznego, wykorzystania algorytmów grafowych w rozwiązywaniu zagadnień szeregowania.
6. Ogólna definicja problemu szeregowania zadań w systemach obsługi wraz z przykładowymi zastosowaniami tego modelu w rozwiązywaniu zagadnień praktycznych.
 7. Wykorzystanie modeli grafowych do efektywnej reprezentacji problemów szeregowania zadań.
 8. System otwarty: algorytm Gonzaleza-Sahniego i trudność problemów wielomaszynowych, przykład podejścia opartego o programowanie matematyczne.
 9. System przepływowy: algorytm Johnsona i algorytmy heurystyczne dla systemów wielomaszynowych (metoda agregacji), metoda geometryczna.
 10. System ogólny: algorytm Jacksona i uogólniona metoda geometryczna, algorytmy listowe, metoda propagacji ograniczeń.
 11. Nieklasyczne modele systemów obsługi motywowane problemami praktycznymi: czasy przezbrojeń, czasy transportu, bufor, elastyczne systemy obsługi, zadania wieloprocesorowe.
 12. Podstawowe pojęcia i techniki analizy problemów szeregowania w trybie online.
 13. Przykład analizy wybranego problemu szeregowania zadań od sformułowania modelu, przez określenie jego trudności, konstrukcję różnego typu algorytmów do ich weryfikacji eksperymentalnej.

W ramach ćwiczeń laboratoryjnych studenci realizują następujące zadania:

1. Indywidualne zaprojektowanie, zaimplementowanie i weryfikacja w eksperymencie obliczeniowym prostego algorytmu heurystycznego rozwiązującego zadany problem szeregowania zadań (konkurs na najbardziej efektywny algorytm).
2. Indywidualne zaprojektowanie, zaimplementowanie i weryfikacja w eksperymencie obliczeniowym dowolnego algorytmu rozwiązującego zadany problem szeregowania zadań w ramach określonego limitu czasu (konkurs na najbardziej efektywny algorytm).
3. Indywidualne opracowanie generatora instancji testowych (konkurs na najlepszy generator instancji) i weryfikatora uzyskiwanych rozwiązań.
4. Zespołowe (w grupie laboratoryjnej) opracowanie zasad przeprowadzania konkursów na najbardziej efektywny algorytm, przygotowanie środowiska umożliwiającego przeprowadzanie testów algorytmów oraz ustalenie rankingu rozwiązań zaproponowanych przez poszczególnych studentów.
5. Zespołowe (w grupie laboratoryjnej) wykonanie eksperymentów obliczeniowych z wykorzystaniem wygenerowanych instancji, w tym weryfikacja algorytmów własnych i innych członków grupy, oraz analiza uzyskanych wyników.

Treści programowe są przekazywane z użyciem następujących metod dydaktycznych:

1. Wykład ilustrowany prezentacją multimedialną zawierającą omawiane treści programowe, wzbogaconą przykładami.
2. Ćwiczenia laboratoryjne ilustrujące materiał prezentowany podczas wykładu, polegające na analizie i rozwiązywaniu prostych problemów szeregowania zadań, obejmujące samodzielne projektowanie i implementację algorytmów, planowanie i przeprowadzanie eksperymentów obliczeniowych, przygotowanie sprawozdań, dyskusje i ustne prezentacje rozwiązań, negocjacje w szerokiej grupie studenckiej sposobu przygotowania wspólnego środowiska testowego oraz zasad oceny jakości algorytmów.

Literatura podstawowa:

1. Handbook on scheduling: from theory to practice, J. Błażewicz, K. Ecker, E. Pesch, M. Sterna, G. Schmidt, J. Węglarz, Springer, Cham, 2019.
2. Planning and scheduling in manufacturing and services, M. Pinedo, Springer, New York, 2007.
3. Late work scheduling in shop systems, M. Sterna, Rozprawy nr 405, Wydawnictwo PP, Poznań, 2006.
4. Badania operacyjne dla informatyków, J. Błażewicz, W. Cellary, R. Słowiński, J. Węglarz, WNT, Warszawa, 1983.

Literatura uzupełniająca:

1. Scheduling: theory, algorithms, and systems, M. Pinedo, Springer, New York, 2016.
2. Scheduling algorithms, P. Brückner, Springer, Berlin, 2007.
3. Handbook of scheduling: algorithms, models, and performance analysis, J.Y.-T. Leung (eds.), CRC Press, Boca Raton, 2004.
4. Operations Research. A Practical Introduction, M.W. Carter, C.C. Price, CRC Press, Boca Raton, 2001.
5. Handbook of combinatorial optimization, D.-Z. Du, P.M. Pardalos (eds.), Kluwer Academic Publishers, Boston, 1998.
6. The disjunctive graph machine representation of the job shop scheduling problem, J. Błażewicz, E. Pesch, M. Sterna, European Journal of Operational Research 127/2 (2000), 317-331 ([https://doi.org/10.1016/S0377-2217\(99\)00486-5](https://doi.org/10.1016/S0377-2217(99)00486-5))
7. Open shop scheduling problems with late work criteria, J. Błażewicz, E. Pesch, M. Sterna, F. Werner, Discrete Applied Mathematics 134 (2004), 1-24 ([https://doi.org/10.1016/S0166-218X\(03\)00339-1](https://doi.org/10.1016/S0166-218X(03)00339-1))
8. Scheduling on parallel identical machines with late work criterion. Offline and online cases. X. Chen, M. Sterna, X. Han, J. Błażewicz, Journal of Scheduling 19/6 (2016), 729-736 (<https://doi.org/10.1007/s10951-015-0464-7>)
9. A survey of scheduling problems with late work criteria, M. Sterna, Omega 39/2 (2011), 120-129 (<https://doi.org/10.1016/j.omega.2010.06.006>)
10. Polynomial time approximation scheme for two parallel machines scheduling with a common due date to maximize early work, M. Sterna, K. Czerniachowska, Journal of Optimization Theory and Applications 174/3 (2017), 927-944, (<https://doi.org/10.1007/s10957-017-1147-7>)

Bilans nakładu pracy przeciętnego studenta		
Czynność	Czas (godz.)	
1. Udział w wykładach: 15 x 2 godz.	30	
2. Udział w laboratoriach: 15 x 2 godz.	30	
3. Przygotowanie do zajęć laboratoryjnych: 15 x 2 godz.	30	
4. Udział w konsultacjach związanych z realizacją procesu kształcenia	2	
5. Zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu = 1 godz.)	8	
6. Przygotowanie do zaliczenia kolokwium podczas wykładu	12	
7. Przygotowanie do zaliczenia zadań laboratoryjnych	8	
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	120	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	62	2
Zajęcia o charakterze praktycznym	68	2