

Egzamin licencjacki/inżynierski

25 czerwca 2021

Informacja dla zdających egzamin na kierunku informatyka

Z sześciu poniższych zestawów zadań (Matematyka I, Programowanie, Matematyka dyskretna, Algorytmy i struktury danych, Metody numeryczne, Matematyka II) należy wybrać i przedstawić na osobnych kartkach rozwiązania trzech zestawów.

Informacja dla zdających egzamin na kierunku ISIM

Z sześciu poniższych zestawów zadań (Matematyka I, Programowanie, Matematyka dyskretna, Algorytmy i struktury danych, Metody numeryczne, Języki formalne i złożoność obliczeniowa) należy wybrać i przedstawić na osobnych kartkach rozwiązania trzech zestawów.

Informacja dla wszystkich zdających

Za brakujące (do trzech) zestawy zostanie wystawiona ocena niedostateczna z urzędu. Egzamin uważa się za zaliczony, jeśli student rozwiąże z oceną dostateczną co najmniej 2 zestawy. Wtedy ocena z egzaminu jest średnią arytmetyczną ocen z trzech wybranych zestawów. Na rozwiązanie przeznaczona jest czas $3 \times 40 = 120$ minut. Po wyjściu z sali egzaminacyjnej w czasie egzaminu nie ma możliwości powrotu do tej sali i kontynuowania pisania egzaminu.

Matematyka I — Logika dla informatyków

W tym zadaniu chcemy sprawdzić, czy relacja *bardziej ogólne* w zbiorze podstawień jest relacją porządku. Dla przypomnienia: przez podstawienie w zbiorze termów $T(\Sigma, V)$ nad niepustą sygnaturą Σ i nieskończonym zbiorem zmiennych V rozumiemy dowolną funkcję $\sigma : V \rightarrow T(\Sigma, V)$, którą w naturalny sposób rozszerzamy do funkcji (zwyczajowo oznaczanej tym samym symbolem) $\sigma : T(\Sigma, V) \rightarrow T(\Sigma, V)$. Mówimy, że podstawienie σ jest bardziej ogólne od podstawienia θ i piszemy $\sigma \preceq \theta$, jeśli istnieje takie podstawienie ρ , że $\theta = \rho \cdot \sigma$, gdzie \cdot oznacza składanie funkcji.

Czy relacja \preceq jest

- a) zwrotna ?
- b) słabo antysymetryczna?
- c) przechodnia?

Jeśli relacja \preceq jest porządkiem, to czy ma on

- d) elementy minimalne?
- e) elementy maksymalne?

Wszystkie odpowiedzi należy uzasadnić.

Matematyka II — Algebra

Zadanie 1. (6 punktów) Niech A będzie macierzą nieosobliwą, $S = AA^T$. Wykazać, że S jest macierzą dodatnio określoną.

Zadanie 2. (4 punkty) λ jest wartością własną macierzy A . Wskazać jedną z wartości własnych macierzy A^2 . Jakie są wartości własne macierzy $M = a_0I + a_1A + \dots + a_nA^n$ w porównaniu z tymi macierzy A ?

Zadanie 3. (4 punkty) Wykazać, że $\langle 2\mathbb{Z}, + \rangle$ (zbiór liczb całkowitych, parzystych wraz z dodawaniem) jest podgrupą grupy $\langle \mathbb{Z}, + \rangle$.

Progi punktowe: 4, 6, 8, 10, 12 punktów.

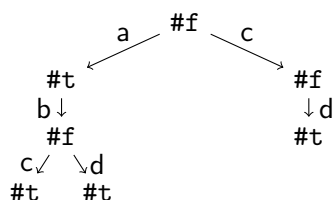
Programowanie – wariant A

Za tę część egzaminu można otrzymać 20 punktów. Aby otrzymać ocenę dostateczną, należy zdobyć 7 punktów, próg dla dst+ to 9p, dla db – 11p, dla db+ 13p, dla bdb – 15p.

Drzewa prefiksowe

Drzewa prefiksowe są strukturami danych służącymi do reprezentacji zbiorów ciągów elementów (nad pewnym alfabetem). Idea jest prosta: krawędzie drzewa są etykietowane elementami naszego alfabetu, zaś wierzchołki zawierają informację, czy ścieżka od korzenia do danego wierzchołka należy do zbioru reprezentowanego przez nasze drzewo.

Przykładowo, poniższe drzewo reprezentuje zbiór ciągów liter $\{a, abc, abd, cd\}$. Zwróćmy uwagę, że ciąg pusty nie należy do drzewa, podobnie jak ciągi ab czy c , mimo że odpowiednie ścieżki są w drzewie obecne — wynika to z oznaczeń w odpowiednich węzłach.



Ponieważ drzewa prefiksowe służą do reprezentacji zbiorów skończonych (o szczególnym kształcie elementów), powinny dostarczać standardowych operacji na zbiorach: sprawdzania należenia do zbioru, dodawania czy usuwania elementów, sprawdzania równości, itp.

Zadanie 1. (6p.) Zaproponuj reprezentację drzew prefiksowych w wybranym języku funkcyjnym (Haskell, Ocaml, Racket). Wykorzystaj narzędzia abstrakcji danych właściwe dla wybranego języka, ale jeśli chcesz użyć bibliotecznej struktury danych, przytocz jej definicję. Następnie opisz niezmienniki swojej struktury danych przy pomocy funkcji, która rozstrzyga czy otrzymane drzewo prefiksowe jest poprawnie sformowane.

Zadanie 2. (6p.) Jak powyżej, ale w wybranym języku imperatywnym (C, Python, Java).

Zadanie 3. (8p.) Dla wybranej z powyższych reprezentacji zaimplementuj następujące funkcje na drzewach prefiksowych:

- **lookup**, rozstrzygającą czy dany ciąg należy do zbioru reprezentowanego przez drzewo;
- **insert**, dodający dany ciąg do zbioru reprezentowanego przez drzewo;
- **remove**, usuwający dany ciąg ze zbioru reprezentowanego przez drzewo;
- **equal**, rozstrzygającą czy dwa drzewa reprezentują te same zbiory ciągów.

Oczywiście jeśli piszesz w języku funkcyjnym, powstrzymaj się od używania jego cech imperatywnych i *vice versa*.

Programowanie – wariant B

Za zadania można otrzymać 20 punktów. Aby otrzymać ocenę dostateczną, należy zdobyć 7 punktów, próg dla dst+ to 9p, dla db – 11p, dla db+ 13p, dla bdb – 15p.

Zadanie 1. Gramatyka G_1 z symbolem startowym S nad alfabetem $\{a, b, c, d\}$ dana jest za pomocą następującego zbioru produkcji:

$$X \rightarrow aXb, X \rightarrow ab, Y \rightarrow cYd, Y \rightarrow cd, S \rightarrow XSY, S \rightarrow XY$$

Dla gramatyki G przez $L(G)$ rozumiemy język generowany przez G . Dla wyrażenia regularnego r przez $\mathcal{L}(r)$ rozumiemy język opisany przez wyrażenie r .

- a) Czy $aabbcd$ należy do $L(G_1)$? Odpowiedź uzasadnij. **(1p)**
- b) Czy gramatyka G_1 jest jednoznaczna? Odpowiedź uzasadnij. **(3p)**
- c) Przedstaw wyrażenie regularne (o ile to możliwe) lub gramatykę bezkontekstową (w przeciwnym przypadku) generującą zbiór $A_1 = L(G_1) \cap \mathcal{L}(a^*b^*c^*d^*)$. Odpowiedź uzasadnij. **(2p)**
- d) Napisz w języku imperatywnym funkcję, która bierze jako wejście napis i zwraca wartość logiczną, równą True wtedy i tylko wtedy, gdy ten napis należy do zbioru $L(G_1) \cap \mathcal{L}((ab+cd)^*)$. Możesz używać języka wybranego z następującej listy: C, C++, Java, C#, Python, Ruby, Rust, Go **(4p)**

Zadanie 2. Napisz w Prologu predykat `generate_expr(+N, +Val, -E)`, który jako argument bierze liczbę naturalną N oraz liczbę naturalną Val i jest prawdziwy, gdy zachodzi koniunkcja następujących warunków:

- i) E jest wyrażeniem zbudowanym ze znaków dodawania, mnożenia, odejmowania oraz cyfr,
- ii) E ma co najwyżej N operatorów
- iii) wartością E jest Val .

Predykat powinno dać się użyć do generowania takich wyrażeń. Możesz tworzyć predykaty pomocnicze, należy jednak wyraźnie napisać, co taki predykat robi i jakie znaczenie mają jego poszczególne argumenty. **(5p)**

Zadanie 3. Niech F będzie funkcją, która dla liczby naturalnej n (zapisanej w systemie dziesiętnym) oblicza jej sumę cyfr. Rozważmy proces, w którym liczymy wartość F dla liczby naturalnej, potem liczymy wartość F dla wyniku i powtarzamy to aż do momentu, w którym kolejne aplikacje „nic nie dają” (bo otrzymamy liczbę jednocyfrową równą swojej sumie cyfr). Tę właśnie liczbę nazwiemy *granica cyfrową* liczby n .

Napisz w Haskellu lub OCamlu funkcję obliczającą granicę cyfrową liczby naturalnej. Możesz definiować funkcje pomocnicze. Dla każdej funkcji pomocniczej opisz znaczenie jej argumentów i wyniku. **(5p)**

Matematyka dyskretna

Podgraf indukowany grafu G składa się z pewnego podzbioru wierzchołków $V' \subseteq V(G)$ i zbioru wszystkich krawędzi, które w G łączą wierzchołki z V' . Pokaż, że jeśli graf zawiera cykl długości nieparzystej, to zawiera podgraf indukowany będący cyklem długości nieparzystej. Czy podobne stwierdzenie jest prawdziwe dla cykli długości parzystej?

Metody numeryczne

Za rozwiązanie zadań można otrzymać łącznie 12 punktów. Otrzymanie 4 pkt. gwarantuje ocenę dostateczną, próg dla dst+ to 5.5 pkt., dla db – 7 pkt., dla db+ 8.5 pkt., a dla bdb – 10 pkt.

1. **4 punkty** Dla jakich $x \in \mathbb{R}$ obliczanie z wykorzystaniem arytmetyki zmiennopozycyjnej wartości wyrażenia $(\sin x - x)/x^3$ może wiązać się z utratą cyfr znaczących wyniku? Odpowiedź uzasadnij. Następnie zaproponuj sposób obliczenia w arytmetyce fl wyniku dokładniejszego.
2. **4 punkty** (a) Podaj definicję naturalnej funkcji sklejaney trzeciego stopnia interpolującej funkcję f w węzłach $x_0 < x_1 < \dots < x_n$. Gdzie funkcje tego typu znajdują zastosowania? (b) Czy $f(x) = x^3$ jest naturalną interpolacyjną funkcją sklejaną trzeciego stopnia dla danych

$$\begin{array}{c|c|c|c} x_k & -2 & 0 & 2 \\ \hline y_k & -8 & 0 & 8 \end{array} ?$$

Odpowiedź uzasadnij.

3. **4 punkty** Zaproponuj algorytm odwracania nieosobliwej macierzy trójkątnej dolnej. Podaj jego złożoność obliczeniową i pamięciową.

Algorytmy i struktury danych

Za rozwiązanie obydwu zadań z tej części można otrzymać w sumie do 9 punktów. Skala ocen: poniżej 3 punktów — ocena niedostateczna (egzamin niezdany), 3 punkty dają ocenę dostateczną, 4 — dostateczną z plusem, 5 — dobrą, 6 — dobrą z plusem, 7 albo więcej punktów daje ocenę bardzo dobrą.

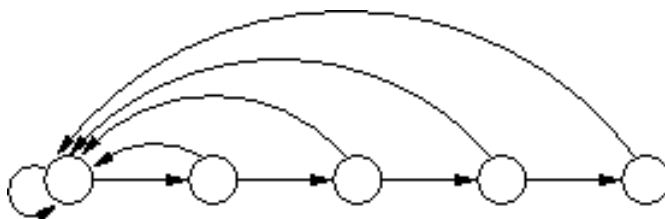
Zadanie 1: proces technologiczny z minimalną liczbą procesorów (4 punkty)

Pewien proces technologiczny realizujący jakieś zadanie został zaprojektowany w taki sposób, że zaplanowano w nim n niezależnie wykonujących się procesów. Każdy proces rozpoczyna się o określonej godzinie t_i i trwa określoną liczbę minut d_i , dla $i = 1 \dots n$. Do prawidłowego wykonania całego zadania każdemu procesowi należy przydzielić procesor na wyłączność, czyli od momentu t_i do $t_i + d_i$. Trzeba więc wyliczyć jaka jest minimalna liczba procesorów potrzebna do prawidłowej realizacji zadania.

Opracuj efektywną metodę rozwiązującą ten problem. Opisz ideę algorytmu, a potem zapisz go w pseudokodzie (wraz z niezbędnymi komentarzami). Uzasadnij, że opisany algorytm działa poprawnie oraz oszacuj jego złożoność czasową.

Zadanie 2: listowa reprezentacja zbiorów rozłącznych (5 punktów)

Rozważmy listową implementację struktury danych dla zbiorów rozłącznych. Każdy zbiór jest reprezentowany jako lista swoich elementów. Reprezentantem zbioru jest pierwszy element na liście. Każdy element oprócz wskaźnika do następnego węzła ma dodatkowo bezpośredni wskaźnik do reprezentanta (reprezentant zbioru, czyli głowa listy, ma ten wskaźnik ustawiony na siebie). Dzięki wskaźnikom powrotnym koszt operacji *Find* jest stały $O(1)$. Operacja *Init* jest również bardzo prosta i polega na utworzeniu n jednoelementowych list, co zajmuje czas liniowy $O(n)$, gdzie n to liczba wszystkich elementów.



Operacja *Union* też wydaje się prosta: połączenie zbiorów sprowadza się do połączenia list — wprowadzie koszt połączenia dwóch list jest stały (wymaga to pamiętania wskaźnika do ostatniego elementu), ale należy jeszcze we wszystkich elementach listy dołączanej uaktualnić wskaźnik na reprezentanta, co zajmuje czas liniowy względem jej długości. Prosta heurystyka, która pozwala zmniejszyć koszt operacji *Union* (w sensie zamortyzowanym) polega na dołączaniu krótszej listy na koniec dłuższej (wymaga to przechowywania wraz z listą informacji o jej rozmiarze).

Przeanalizuj teraz złożoność czasową wykonania ciągu m operacji *Union* i *Find*, przy czym w ciągu tym znajduje się $n - 1$ operacji *Union*. Wykaż, że zamortyzowany koszt wykonania na początku operacji *Init*, a potem m operacji *Union* i *Find* wynosi $O(m + n \log n)$.

Druga część zadania polega na takim zmodyfikowaniu struktury listowej dla zbiorów rozłącznych, aby można było wykonać na niej operację *Extract*, która odłącza element od zbioru, do którego on należy i tworzy nowy jednoelementowy zbiór. Operacja ta powinna działać w czasie stałym $O(1)$.

Języki formalne i złożoność obliczeniowa

Przypomnijmy, że:

- Literał to zmienna lub zanegowana zmienna.
- Klauzula to alternatywa literałów.
- Klauzula jest *hornowska*, jeśli co najwyżej jeden z występujących w niej literałów jest niezanegowaną zmienną.

Rozpatrzmy następujące problemy:

HORN-SAT	
Wejście	Formuła boolowska będąca koniunkcją klauzul, spośród których każda jest hornowska
Pytanie	Czy ta formuła jest spełnialna?

2-SAT	
Wejście	Formuła boolowska będąca koniunkcją klauzul, spośród których każda jest alternatywą co najwyżej dwóch literałów
Pytanie	Czy ta formuła jest spełnialna?

SAT ₂	
Wejście	Formuła boolowska będąca koniunkcją klauzul taka, że każda zmienna występuje w niej co najwyżej dwa razy
Pytanie	Czy ta formuła jest spełnialna?

HORN+2-SAT	
Wejście	Formuła boolowska będąca koniunkcją klauzul, spośród których każda jest albo hornowska, albo zawiera co najwyżej dwa literały
Pytanie	Czy ta formuła jest spełnialna?

HORN-SAT _{albo-2}	
Wejście	Formuła boolowska będąca koniunkcją klauzul takich, że każda zmienna, która występuje w klauzuli, która nie jest hornowska, ma co najwyżej dwa wystąpienia.
Pytanie	Czy ta formuła jest spełnialna?

1. (1 punkt) Wybierz jeden z problemów HORN-SAT, 2-SAT, SAT₂ i przypomnij, jaka jest jego złożoność, oraz podaj główną ideę dowodu tej złożoności.
2. (2 punkty) Jaka jest złożoność problemu HORN+2-SAT?
3. (2 punkty) Jaka jest złożoność problemu HORN-SAT_{albo-2}?

Ocena to liczba zdobytych punktów.