Warsaw Center
of Mathematics
and Computer Science

21 maja 2015

egzamin kwalifikacyjny na studia doktoranckie ze stypendium w instytucjach WCNM

---

kod . . . .

imię i nazwisko . . . .

kierunek . . . .

Kandydat wybiera **5 zadań** i odnotowuje swój wybór na tej karcie. Jeśli zaznaczono więcej niż 5 zadań, tylko pierwszych 5 zostanie ocenione.

Rozwiązanie każdego zadania musi być zapisane **na osobnej kartce**.

Każda kartka z rozwiązaniem musi być podpisana **czterocyfrowym kodem** podanym powyżej. Rozwiązań **nie należy** podpisywać imieniem i nazwiskiem.

Wybrane zadania:

1. ☐ 2. ☐ 3. ☐ 4. ☐ 5. ☐

6. ☐ 7. ☐ 8. ☐ 9. ☐ 10. ☐

11. ☐ 12. ☐ 13. ☐ 14. ☐ 15. ☐

Podpis: . . . .

Zadanie 1. Rozwiąż następujące zagadnienie optymalizacyjne

$$\text{zminimalizuj} \quad -\sum_{i=1}^{n} \log(\alpha_i + x_i)$$

przy warunkach dla każdego $i = 1, ..., n$, $x_i \geq 0$,

$$(1, ..., 1)^T (x_1, ..., x_n) = 1,$$

wiedząc, że $\alpha_i > 0$ dla każdego $i = 1, ..., n$.

Zadanie 2. Znajdź liczbę podzbiorów zbioru $\{1, \ldots, n\}$, które nie zawierają dwóch kolejnych elementów.

Zadanie 3. Niech p będzie liczbą pierwszą różną od 2, i niech $a \in \mathbb{Z}_p^*$. Rozpatrzmy permutację $\sigma_a : \mathbb{Z}_p^* \ni b \mapsto ab \in \mathbb{Z}_p^*$. Pokazać, że $a^{(p-1)/2} = sgn(\sigma_a)$, gdzie sgn oznacza znak permutacji.

Zadanie 4. Niech l^∞ będzie przestrzenią ciągów ograniczonych o wyrazach z $\mathbb{R}$ z normą sup. Dla ustalonego ciągu $\{a_1, a_2, \ldots\}$ określamy $P\{a_1, \ldots\} = \{a_1/2, a_2/4, \ldots, a_i/2^i, \ldots\}$, $P\colon l^\infty \to l^\infty$.

Wykaż, że P jest operatorem zwartym tj. jeśli ciąg ciągów $\{f_n\} \subset l^\infty$ zawarty jest w pewnej kuli w l^∞ to $\{Pf_n\}$ zawiera podciąg zbieżny.

Zadanie 5. Do macierzy

$$A = \begin{bmatrix} 3/2 & 1/2 & 0 & 0 \\ 1/2 & 3/2 & 0 & 0 \\ 0 & 0 & 0 & -2 \\ 0 & 0 & -2 & 0 \end{bmatrix}$$

zastosowano odwrotną metodę potęgową

$$\vec{y}_k := (A - \sigma I)^{-1} \vec{x}_{k-1}, \qquad \vec{x}_k := \vec{y}_k / \|\vec{y}_k\|_2, \qquad k = 1, 2, \ldots$$

z wektorem początkowym $\vec{x}_0 = [1, 0, 0, 1]^T$. Dla jakich wartości parametru $\sigma \in \mathbb{R}$

(a) ciąg $\{\vec{x}_k\}_{k=0}^{\infty}$ jest zbieżny do pewnego wektora własnego macierzy A?

(b) ciąg $\{\eta_k\}_{k=0}^{\infty}$, gdzie $\eta_k = \vec{x}_k^T A \vec{x}_k$, jest zbieżny do pewnej wartości własnej macierzy A?

Zadanie 6. Niech $X_1, ..., X_n$ będzie próbką z rozkładu o gęstości

$$f_\theta(x) = \begin{cases} \frac{\theta}{(1-x)^2} \cdot e^{-\theta x/(1-x)} & \text{dla } 0 < x < 1 \\ 0 & w\ p.p. \end{cases}$$

gdzie $\theta > 0$ jest nieznanym parametrem. Niech $\theta_0 > 0$ będzie daną liczbą. Skonstruować test hipotezy $H_0 : \theta \geq \theta_0$ przeciwko alternatywie $H_0 : \theta < \theta_0$, który jest jednostajnie najmocniejszy na poziomie istotności α.

(Kwadryle typowych rozkładów prawdopodobieństwa uznajemy za znane.)

Zadanie 7. Dana jest funkcja ciągła $f : \mathbb{R} \to \mathbb{R}$, $f(0) = 0$. Rozpatrzmy zagadnienie Cauchy'ego

$$\frac{dx}{dt} = f(x), \qquad x(0) = x_0.$$

(a) Załóżmy, że f jest klasy $C^1(0;\infty)$, $f(x) > 0$ dla $x > 0$, $\lim_{x\to 0^+} f'(x) = +\infty$ oraz istnieje funkcja pierwotna F(x) funkcji $\frac{1}{f(x)}$ i $F(0) = 0$.

Rozwiąż, zagadnienie Cauchy'ego dla $x_0 = 0$ przy powyższych założeniach. Rozwiązanie uzasadnij.

(b) Niech f będzie funkcją klasy $C^1(\mathbb{R})$, wklęsłą, unimodalną, osiągającą swoje jedyne maksimum w pewnym punkcie $x = a > 0$. Narysować i omówić przebieg rozwiązań zagadnienia dla dowolnego $x \geq 0$ i $t \geq 0$.

Zadanie 8. Niech $X = (X_1, X_2)$ będzie dwuwymiarowym wektorem losowym o rozkładzie $N(0, I_2)$, gdzie I_2 jest 2×2 macierzą identycznościową. Policz prawdopodobieństwo $\mathbb{P}[|X_1| + |X_2| \leq 1]$.

Zadanie 9. Znaleźć algorytm rozwiązujący następujący problem decyzyjny w czasie liniowym.

> *Wejście:* Graf nieskierowany planarny G.
> *Pytanie:* Czy graf G zawiera (jako podgraf) klikę czteroelementową?

Zadanie 10. Dany jest graf $G = (\{1, \ldots, n\}, E)$ wraz z funkcją $w : E \to \mathbb{R}_+$, dwa różne wierzchołki $s, t \in V$, oraz liczba $k \in \mathbb{N}_+$. Długością ścieżki w G jest suma wartości w dla jej krawędzi. Ścieżki można porównywać leksykograficznie traktując je jako ciągi wierzchołków. Zaproponuj efektywny algorytm podający k-tą ścieżkę w porządku leksykograficznym na zbiorze najkrótszych ścieżek z s do t, lub stwierdzający, że taka ścieżka nie istnieje.

Zadanie 11. Jaka jest złożoność obliczeniowa następującego problemu decyzyjnego?

> *Wejście:* Wyrażenie regularne E i liczba naturalna n, reprezentowana binarnie.
> *Pytanie:* Czy $L(E)$ zawiera przynajmniej jedno słowo długości n?

Zadanie 12. Dla języka L, niech *pół*(L) oznacza język $\{w : ww \in L\}$. Czy następujący problem jest rozstrzygalny?

> *Wejście:* język bezkontekstowy L.
> *Pytanie:* Czy *pół*(L) jest niepusty?

Zadanie 13. Czy da się wyrazić w logice pierwszego rzędu, że graf nieskierowany jest regularny?

Przypomnijmy, że graf nieskierowany jest *regularny*, jeśli każdy wierzchołek ma ten sam stopień. Graf reprezentujemy jako strukturę relacyjną $\langle V, E\rangle$, gdzie $E \subseteq V \times V$.

Zadanie 14. Powiemy, że ciąg liczb a o długości n *pokrywa* ciąg b, gdy dla każdego i od 1 do n suma i pierwszych elementów a jest nie mniejsza niż suma i pierwszych elementów b.

Poniżej znajduje się przykładowa, być może błędna, implementacja funkcji `pokrywa`, która stwierdza czy a pokrywa b. Znajdź i popraw błąd w implementacji, a następnie zweryfikuj za pomocą logiki Hoare'a funkcję `pokrywa`.

```
int pokrywa(int[] a, int la, int[] b, int lb) {
    int s = 0;

    for(int i=0; i<la; i++) {
        s += a[i] - b[i];
        if (s<=0) return 0;
    }
    return 1;
}
```

Zadanie 15. Rozważmy system przeznaczony do analizy podejrzanej aktywności w sieci. Działa on na wielo-procesorowych ruterach i ma za zadanie wybiórcze monitorowanie ruchu sieciowego. Liczne wątki tworzące system realizują to następująco. Każdy przychodzący pakiet jest błyskawicznie klasyfikowany do tak zwanego strumienia przy użyciu wyspecjalizowanych układów sprzętowych. Od czasu do czasu, dla każdego strumienia, system wybiera losowy pakiet, który jest poddawany bardziej szczegółowej i kosztownej obliczeniowo analizie, na podstawie której generowane są alarmy dla administratora sieci.

Twoim zadaniem jest zaimplementowanie wyboru losowego pakietu ze strumienia i udowodnienie poprawności swojego rozwiązania. Celem rozwiązania powinno być zapewnienie jak największej wydajności i skalowalności systemu jeśli chodzi o liczbę strumieni, S. Dokładniej, należy zaimplementować:

Strukturę/klasę `Stream` Implementacja powinna specyfikować wszystkie oraz jedynie te pola `Stream`, które opisują struktury danych niezbędne do realizacji zadania.

Funkcję/metodę `void packetClassified(Stream * s, Packet * p)` Funkcja ta jest wołana w momencie, gdy ruter odebrał pakiet `p` i zaklasyfikował go do strumienia `s` (obiekty `s` i `p` są alokowane i automatycznie odśmiecane przez ruter). Może być ona wołana niezależnie i równolegle przez wiele wątków, reprezentujących na przykład różne interfejsy sieciowe rutera. Nie wiadomo nic więcej odnośnie rozkładu wywołań tej funkcji w czasie i pomiędzy strumieniami.

Funkcję/metodę `Packet * getRandomPacket(Stream * s)` Funkcja ta jest wołana przez ruter w momencie, gdy potrzebny jest losowy pakiet ze strumienia `s` do dokładnej analizy. Ma zwrócić pakiet wybrany losowo z rozkładem jednostajnym spośród wszystkich pakietów zaklasyfikowanych do strumienia `s` w okresie albo od poprzedniego wywołania tej funkcji dla `s`, jeśli takie było, albo od początku działania systemu, jeśli funkcja nie była jeszcze wołana dla strumienia `s`. Jeśli w tym okresie żadne pakiety nie zostały zaklasyfikowane do `s`, funkcja ma zwrócić `null`. Może być ona wołana niezależnie i równolegle przez wiele wątków. Nie wiadomo nic więcej odnośnie rozkładu wywołań tej funkcji w czasie i pomiędzy strumieniami.

Funkcję/konstruktor `void initStream(Stream * s)` Funkcja ta jest wołana w momencie uruchomienia rutera, zanim ruter zacznie przyjmować jakiekolwiek pakiety. Ma za zadanie zainicjalizować struktury danych dla strumienia `s`.

Do zarządzania czasem życia pakietów należy skorzystać z funkcji/metod `void pinPacket(Packet * p)` oraz `void unpinPacket(Packet * p)`. Obie funkcje są bezpieczne jeśli chodzi o równoległe wywołania przez różne wątki. Pierwsza z nich zaznacza, że pakiet nie będzie odśmiecony. W szczególności, każdy pakiet, dla którego nie zostanie wywołane `pinPacket` wewnątrz funkcji `packetClassified` zostanie odśmiecony po powrocie z tej funkcji. Analogicznie, druga z nich odznacza pakiet, sprawiając, że staje się on kandydatem do odśmiecenia. Można założyć, że funkcja analizująca losowe pakiety woła `unpinPacket` po zakończeniu analizy każdego pakietu zwróconego przez `getRandomPacket`.

Do generowania liczb losowych należy użyć sprzętowego generatora liczb pseudolosowych rutera, dostępnego jako funkcja `double hwRandom()`. Funkcja ta zwraca liczbę pseudolosową z rozkładem jednostajnym z przedziału $[0 \ldots 1)$. Jest bezpieczna jeśli chodzi o równoległe wywołania przez różne wątki. Można także założyć, że operacje na `double` nie są obarczone błędami zaokrągleń.

Do zarządzania współbieżnością można użyć dowolnych standardowych mechanizmów synchronizacyjnych dostępnych z przestrzeni użytkownika, z tym że należy wyjaśnić właściwości, jakie te mechanizmy powinny spełniać i uzasadnić wybór konkretnych mechanizmów. Można przyjąć, że zaimplementowane funkcje będą wywoływane przez niezmieniającą się liczbę wątków, P, odpowiadającą z grubsza liczbie dostępnych na ruterze procesorów.