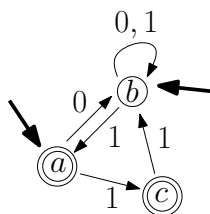


Bakalářské zkoušky (příklady otázek)

léto 2018

1 Automaty (3 body)

1. Převeďte nedeterministický konečný automat $A = (\{a, b, c\}, \{0, 1\}, \delta, \{a, b\}, \{a, c\})$ z obrázku na ekvivalentní deterministický automat.
2. Do jakých tříd jazyků v Chomského hierarchii patří jazyk $L(A)$?



2 Algoritmy a datové struktury: vyhledávání v textu (3 body)

1. Definujte vyhledávací automat algoritmu KMP (Knuth-Morris-Pratt). Jak vypadají jeho stavy, dopředné a zpětné hrany?
2. Nakreslete vyhledávací automat pro slovo KOCKODAN.
3. Jakou časovou složitost má konstrukce automatu a jakou jeho použití k vyhledání všech výskytů slova v textu?

3 Databáze (3 body)

1. Načrtněte, jak byste v databázové tabulce reprezentovali binární vztah M:N. Lze z definice tabulky $T(\underline{FK}_1, \underline{FK}_2)$, vzniklé převodem binárního vztahu, určit jeho původní kardinalitu (1:1, 1:N, M:N)? Vysvětlete.
2. Uvažujte transakce $T_1: W(A) R(B) W(C)$ a $T_2: R(A) W(B) W(C)$. Je rozvrh $S: W_1(A) R_2(A) R_1(B) W_2(B) W_1(C) W_2(C)$ konfliktově serializovatelný (conflict serializable)? Vysvětlete proč a pokud není, navrhněte úpravu, aby byl.

4 Objektový interface pro REST API server (3 body)

Mějme specializovaný web server pro REST API implementovaný v mainstreamovém objektově orientovaném staticky typovaném jazyce (C++, C# nebo Java) s následujícími vlastnostmi. Server umožňuje, aby se do něho dynamicky vkládaly služby, přičemž služba je objekt třídy implementující rozhraní `IService`, který zapouzdřuje několik REST metod. Služby jsou identifikovány řetězci, které musí být unikátní v rámci serveru, REST metody jsou rovněž identifikovány řetězci, které musí být unikátní v rámci dané služby. Součástí rozhraní třídy `IService` je i schopnost publikovat seznam svých REST metod. Souvislost mezi REST metodami a HTTP metodou volání (GET, POST) neřešíme, v dalším kontextu je metoda bez přívlastku chápána jako členská funkce třídy.

Zpracování požadavku přes REST API probíhá tak, že server přečte URL a další data z HTTP protokolu a z nich dekoduje identifikátor služby, identifikátor REST metody a kolekci parametrů. Dle identifikátorů najde ve vnitřních záznamech požadovanou službu a na ní zavolá metodu implementující požadovanou REST metodu, přičemž metoda dostane kolekci parametrů jako vstupní argument a vrátí řetězec. Pro naše účely si představme kolekci parametrů jako mapu/slovník (konkrétní typ si upravte dle zvoleného jazyka), kde klíče jsou řetězce (názvy parametrů) a hodnoty jsou objekty typu `ParameterValue` (další detaily jsou pro naši úlohu nezájímavé).

Server je reprezentován objektem třídy `Server`. Tato třída má (mimo jiné) dvě podstatné metody, které by mohly být symbolicky zapsány přibližně takto (přesný zápis závisí na použitém jazyce):

```
public void add(IService service)
private string dispatch(string serviceId, string methodId, ParametersCollection parameters)
```

Metoda `add` zaregistruje novou službu v rámci serveru. Tato metoda může být relativně pomalá a volá se typicky jen při startu aplikace (např. když jsou načítány zásuvné moduly). Metoda `dispatch` najde a zavolá cílovou metodu příslušné služby a je volána výhradně z vnitřní implementace serveru při zpracování požadavku ze sítě. Metoda `dispatch` by neměla mít velký overhead.

1. Navrhnete, jak by měl vypadat interface `IService` a stručně (např. komentářem) vysvětlíte význam jednotlivých metod (pokud není naprosto zřejmý z názvu metody).
2. Představte příklad konkrétní třídy implementující rozhraní `IService`. Zaměřte se zejména na realizaci části rozhraní zodpovědné za předání informací o nabízených metodách.
3. Stručně (případně v pseudokódu) popište, jak bude vypadat implementace metod `add` a `dispatch` třídy `Server`. Pokud tyto metody potřebují přistupovat ke členským proměnným třídy, popište tyto proměnné také. Připomeňme, že metoda `dispatch` by měla být rozumně efektivní.

Drobné chyby v syntaxi budou tolerovány, avšak celkový návrh a logika rozhraní by měly obecně odpovídat principům a zvyklostem zvoleného jazyka. V jazycích, které to umožňují, je povoleno rozumné použití reflexe, nicméně reflexe není nutně vyžadována.

5 Souborový systém (3 body)

Tato otázka obsahuje zjednodušený popis souborového systému FAT. Pokud chcete, můžete v odpovědi uvažovat i skutečný souborový systém FAT (pokud ano, výslovně to napište).

Oddíl naformátovaný (zjednodušeným) souborovým systémem FAT12/16/32 je rozdělený na 3 části:

1. boot sektor (nultý logický sektor oddílu),
2. tzv. tabulka FAT, viz dále (1. až N. sektor oddílu),
3. datové sektory (N+1. sektor až poslední sektor oddílu), které obsahují samotná data souborů (nebo adresářů, nicméně jelikož ty se z pohledu alokace sektorů od souborů v souborovém systému neliší, tak se jimi dále nebudeme zabývat zvlášť). Pro jednoduchost předpokládejme, že alokační jednotka souborového systému je právě jeden sektor disku. Sektory disku mají jednu z obvyklých velikostí – označme ji jako X bytů.

Každý datový sektor oddílu je přiřazený právě jednomu souboru, nebo je označený jako volný. Každý soubor na disku zabírá určité množství celých sektorů, přičemž ale tyto sektory nemusí být konkrétnímu souboru přiděleny kontinuálně (soubory mohou být na disku fragmentované). V adresářovém záznamu pro konkrétní soubor je zapsáno číslo datového sektoru (číslovány od 1), který obsahuje data prvních X bytů souboru (pro soubory s velikostí menší nebo rovnou X bytů je to zároveň poslední sektor souboru). Pro soubory s velikostí větší než X bytů nalezneme informaci o dalších sektorech přidělených souboru ve FAT tabulce oddílu – pro každý soubor (který zabírá např. M sektorů) obsahuje FAT tabulka „zakódovanou“ obdobu jednosměrně vázaného seznamu posloupnosti čísel 2. až M . sektoru souboru. Přesný obsah FAT tabulky je následující:

Pro každý datový sektor obsahuje FAT tabulka právě jeden Z bitový záznam ($Z = 12$ bitů pro FAT12, 16 bitů pro FAT16, 32 bitů pro FAT32) – tento záznam je bezznaménkové Z bitové celé číslo uložené v pořadí little endian. V tabulce FAT je tedy právě tolik záznamů, kolik oddíl obsahuje datových sektorů, a žádné jiné informace FAT tabulka neobsahuje (FAT tabulka je tedy fakticky jen pole celých čísel). Záznam na offsetu 0 v prvním sektoru FAT tabulky obsahuje informaci o 1. datovém sektoru oddílu, hned za ním (bez paddingu) následuje záznam pro 2. datový sektor oddílu, pak záznam pro 3. datový sektor oddílu, atd. Pokud záznam pro datový sektor A obsahuje číslo 0, tak je tento datový sektor volný a není přidělen žádnému souboru. Pokud záznam pro datový sektor A obsahuje číslo B , tak je datový sektor A přidělený nějakému souboru, a po X bytech dat toho souboru uložených v datovém sektoru A je následujících X bytů souboru uložených v datovém sektoru B (kde

pro fragmentované soubory nemusí být B rovno $A+1$, a obecně může být B být větší i menší než A). Pokud záznam pro datový sektor A obsahuje maximální hodnotu Z bitového čísla, pak je sektor A posledním sektorem nějakého souboru (M . sektorem přiděleným souboru o velikosti M sektorů). Pokud tedy např. data souboru `A.TXT` budou ležet v datových sektorech 10, 7, 8, 15, tak v adresářovém záznamu pro `A.TXT` bude zapsáno číslo 10 a ve FAT tabulce bude v záznamu pro sektor 10 uloženo číslo 7, v záznamu pro sektor 7 číslo 8, v záznamu pro sektor 8 číslo 15, a v záznamu pro sektor 15 maximální hodnota Z bitového čísla.

1. Jaká může být obvyklá hodnota X ? Pokud je velikost oddílu právě 1 GiB, je vhodné tento oddíl naformátovat souborovým systémem FAT16 nebo FAT32? Vysvětlete proč.
2. V takovém 1 GiB oddílu máme uloženo 100 000 souborů, kde každý má velikost 1 KiB. Pokud nyní chceme přečíst obsah všech těchto souborů, bylo by pro typický disk vhodné načíst si předem celou FAT tabulku do paměti? Nebo by bylo vhodnější před čtením každého ze souborů znovu načíst do paměti pouze ty sektory FAT tabulky, o kterých zjistíme, že obsahují informace potřebné pro právě čtený soubor? Vysvětlete proč.
3. Předpokládejte, že v "globální proměnné" (resp. statické proměnné nějaké třídy) `fat` typu pole bytů máme načtené veškeré záznamy celé FAT tabulky oddílu naformátovaného souborovým systémem FAT12 – jeden záznam FAT tabulky má velikost 12 bitů, tedy každé 3 byty pole `fat` obsahují právě 2 záznamy o 2 datových sektorech (záznam pro sektor s nižším číslem je vždy v 1. bytu a ve spodních 4 bitech 2. bytu; záznam pro sektor s vyšším číslem je ve vyšších 4 bitech 2. bytu a ve 3. bytu). V jazyce `C#` nebo Java nebo `C++` naprogramujte proceduru, která jako svůj jediný argument dostane číslo prvního datového sektoru přiděleného nějakému souboru, a má na standardní výstup vypsat čísla všech datových sektorů tomuto souboru přiřazených (dle informací v proměnné `fat`).

6 HTTP protokol (3 body)

Následuje záznam části relace v protokolu HTTP:

```
GET / HTTP/1.1
Host: www.example.org
If-None-Match: "0123456789"

HTTP/1.1 200 OK
Date: Mon, 01 Jan 2000 00:00:00 GMT
Last-Modified: Mon, 01 Jan 2009 00:00:00 GMT
Cache-Control: max-age=3600
ETag: "9876543210"
Accept-Ranges: bytes
Content-Length: 123456
Content-Type: text/html
```

```
<!DOCTYPE html ...
```

1. Jaký je význam prvního řádku dotazu?
2. Proč dotaz obsahuje položku `Host`?
3. Jaké URL zřejmě použil klient v prohlížeči?
4. Jaká další slova mohou zahájit první řádek dotazu a jaký je jejich význam? Nejméně dvě.
5. Jaký význam má položka `Accept-Ranges` v odpovědi?
6. Jaký význam má položka `If-None-Match` v dotazu a jak se používá?
7. Jaký význam má položka `Last-Modified` v odpovědi a jak se používá?

7 Optimalizační metody (3 body)

Nechť $Ax \leq b$ je systém lineárních nerovnic o n proměnných. Vynásobením každé nerovnosti kladnou konstantou můžeme docílit, že první sloupec matice A je vektor obsahující pouze složky 0, -1 and 1. Systém $Ax \leq b$ můžeme tudíž ekvivalentně

zapsat jako

$$\begin{aligned} a'_i x' &\leq b_i & (i = 1, \dots, m_1), \\ -x_1 + a'_j x' &\leq b_j & (j = m_1 + 1, \dots, m_2), \\ x_1 + a'_k x' &\leq b_k & (k = m_2 + 1, \dots, m), \end{aligned}$$

kde $x' = (x_2, \dots, x_n)$ a kde a'_1, \dots, a'_n jsou řádky A bez první složky. Pak se můžeme zbavit x_1 : dokažte, že systém $Ax \leq b$ má řešení právě když systém

$$\begin{aligned} a'_i x' &\leq b_i & (i = 1, \dots, m_1), \\ a'_j x' - b_j &\leq b_k - a'_k x' & (j = m_1 + 1, \dots, m_2, k = m_2 + 1, \dots, m) \end{aligned}$$

má řešení. Ukažte, že opakovaným použitím tohoto kroku je možné vyřešit systém lineárních nerovnic (nebo dokázat, že systém nemá řešení). Jaká bude časová složitost takového postupu?

8 Překladače (3 body)

Níže uvedený kód obsahuje instrukce zásobníkového virtuálního stroje s tímto významem: `iload_N` přidává kopii hodnoty N -té proměnné na vrchol zásobníku, `istore_N` ukládá hodnotu odebranou z vrcholu zásobníku do N -té proměnné, `irem` nahrazuje dvě hodnoty z vrcholu zásobníku jejich celočíselným zbytkem (%), `if_icmpge` je podmíněný skok řízený porovnáním dvou hodnot odebraných z vrcholu zásobníku, `ifgt` je podmíněný skok řízený porovnáním jedné hodnoty s nulou, `goto` je nepodmíněný skok, `ireturn` předává vrchol zásobníku jako návratovou hodnotu funkce.

```
0: iload_1
1: iload_2
2: if_icmpge    22
5: iload_2
6: istore_3
7: iload_1
8: istore_2
9: iload_3
10: istore_1
11: goto        22
14: iload_1
15: iload_2
16: irem
17: istore_3
18: iload_2
19: istore_1
20: iload_3
21: istore_2
22: iload_2
23: ifgt        14
26: iload_1
27: ireturn
```

1. Určete základní bloky a nakreslete control flow graf pro tuto funkci.
2. Které z proměnných jsou (pravděpodobně) parametry funkce?
3. Pokuste se zrekonstruovat původní zdrojový kód v jazyce Java, C# nebo podobném.

9 Hranové obarvení (otázka studijního zaměření – 3 body)

Hranové k -obarvení grafu $G = (V, E)$ je přiřazení čísel (resp. barev) $1, 2, \dots, k$ hranám grafu tak, že žádné dvě hrany se společným vrcholem nemají stejnou barvu. Problém hranového barvení spočívá v nalezení hranového obarvení s nejmenším možným počtem barev.

Navrhňte $(2 - 1/\Delta)$ -aproximační algoritmus pro problém hranového barvení, kde Δ označuje maximální stupeň ve vstupním grafu.

10 Kódy (otázka studijního zaměření – 3 body)

Definujte pojem minimální vzdálenosti pro samoopravné kódy.

Uvažme binární lineární kód $C \subseteq \mathbb{Z}_2^8$ generovaný slovy, která mají právě 2 jedničky na sudých pozicích a právě 2 jedničky na lichých pozicích, tedy

$$C = \text{span}\{(x_1, \dots, x_8) \in \mathbb{Z}_2^8 : x_1 + x_3 + x_5 + x_7 = 2, x_2 + x_4 + x_6 + x_8 = 2\}.$$

(Uvedené sčítání jednotlivých prvků kódového slova je bráno v celých číslech.)

Určete parametry tohoto kódu (délka, velikost, vzdálenost) a sestrojte jeho kontrolní matici.

11 Incidence bodů a přímek (otázka studijního zaměření – 3 body)

Mějme k přirozené číslo a mějme dānu množinu $4k^4$ bodů v rovině

$$B = \{0, 1, \dots, k-1\} \times \{0, 1, \dots, 4k^3-1\}.$$

Dále mějme množinu $4k^5$ přímek P obsahujících všechny přímky s rovnicí $y = ax + b$, kde $a \in \{0, 1, \dots, 2k^2-1\}$ a $b \in \{0, 1, \dots, 2k^3-1\}$.

1. Určete počet incidencí B a P .
2. Může mít systém $4k^4$ bodů a $4k^5$ přímek asymptoticky více incidencí než systém (B, P) výše? (Pokud k odpovědi potřebuje nějaké tvrzení (větu) ze sylabu přednášky Kombinatorická a výpočetní geometrie I, tak toto tvrzení formulujte a odvoďte, jak z něho řešení plyne. Nemusíte ale tvrzení dokazovat.)

12 Mohutnost množin (otázka studijního zaměření – 3 body)

Napište definici “Množina x má menší mohutnost než množina y ”.

Dokažte, že pro každou množinu x platí, že x má menší mohutnost než potenční množina x (v teorii ZF, bez použití axiomu výběru).

13 Množinové systémy (otázka studijního zaměření – 3 body)

1. Necht' A je systém podmnožin množiny $\{1, \dots, n\}$ takový, že každé dvě množiny v A mají neprázdný průnik. Ukažte, že $|A| \leq 2^{n-1}$.
2. Najděte systém podmnožin množiny $\{1, \dots, n\}$ velikosti 2^{n-1} takový, že každé dvě množiny v A mají neprázdný průnik.

14 Matematická lingvistika: formální jazyky a automaty (otázka studijního zaměření – 3 body)

1. Vysvětlete pojem neprojektivity v závislostních stromech.
2. Uveďte příklady dvou neprojektivních vět.

15 Matematická lingvistika: základní formalismy pro popis přirozených jazyků (otázka studijního zaměření – 3 body)

1. Popište základní fakta o tvorbě jazykových korpusů.
2. Rozdělte jazykové korpusy podle typu značkování, pro každý typ uveďte alespoň dva příklady.
3. Vysvětlete pojem valence.

16 Matematická lingvistika: základy teorie informace (otázka studijního zaměření – 3 body)

Házíme třemi pravými mincemi. Pro každou z nich platí, že pravděpodobnost, že padne panna, je $1/2$. Výsledek hodu reprezentujeme jako hodnotu náhodné veličiny $\langle x_1, x_2, x_3 \rangle$, kde $x_i \in \{P, O\}$.

1. Vypočítejte entropii rozdělení této náhodné veličiny.
2. Jak se entropie změní, jestliže právě jedna ze tří mincí bude falešná a padne na ní vždy panna?

17 Naivní Bayesovský klasifikátor (otázka studijního zaměření – 3 body)

1. Popište princip naivního Bayesovského klasifikátoru. Na základě čeho (vzorec) přiřazuje objekty do tříd?
2. Popište význam jednotlivých pravděpodobností v Bayesově vzorci. V čem spočívá naivnost klasifikátoru?
3. Použitím Bayesovského klasifikátoru určete, zda student půjde do školy, jestliže vstane pozdě, nemá oblíbenou přednášku a přší. Pravděpodobnost určete na základě následujících pozorování.

Den	Vstanu	Je oblíbená přednáška ?	Počasí	Jdu do školy
1	Brzo	Ne	Slunce	Ne
2	Normálně	Ne	Děšť	Ne
3	Brzo	Ne	Slunce	Ano
4	Pozdě	Ne	Slunce	Ano
5	Pozdě	Ano	Slunce	Ano
6	Pozdě	Ano	Děšť	Ne
7	Brzo	Ano	Děšť	Ano
8	Normálně	Ne	Slunce	Ne
9	Normálně	Ano	Slunce	Ano
10	Pozdě	Ano	Slunce	Ano
11	Normálně	Ano	Děšť	Ano
12	Brzo	Ne	Děšť	Ano
13	Brzo	Ano	Slunce	Ano

18 Rastrová a vektorová grafika (otázka studijního zaměření – 3 body)

1. Uveďte základní princip rastrové a vektorové grafiky, uveďte příklady grafických formátů, vektorových i rastrových.
2. Jaké jsou výhody vektorové grafiky a pro které operace bychom ji preferovali? Uveďte konkrétní příklady.
3. Naznačte principy komprese rastrových obrázků (nemusíte uvádět detaily).

19 Textury ve 3D grafice (otázka studijního zaměření – 3 body)

1. Jaký je rozdíl mezi 2D a 3D texturou? Uveďte typické aplikace těchto dvou přístupů.

2. Jaké jsou hlavní výhody a nevýhody 3D textur při použití v ray-tracingu?
3. Jak je možné napodobit vnitřní strukturu materiálu (dřevo, mramor) pomocí textury nebo procedurální definice v paraskovém zobrazovači (ray-tracing, path-tracing)? Pozn: Pište o simulaci struktury materiálu, ne o zobrazovači!

20 Flow control a congestion control (otázka studijního zaměření – 3 body)

1. Vysvětlete účel flow control a congestion control v protokolu TCP.
2. Vyjádřete limit přenosové rychlosti vynucený mechanismem flow control v protokolu TCP, pokud znáte maximální velikost flow control okna W a parametry MSS a RTT navázaného spojení.
3. Jaké (nejméně dvě) události během přenosu dat indikují v protokolu TCP congestion ?
4. Uvažujte situaci, kdy více aplikací spuštěných na stejném počítači komunikuje pomocí TCP a kvůli kapacitě odchozí linky jsou omezeny mechanismem congestion control. Získá nějaká aplikace výhodu, pokud si v takové situaci otevře další spojení ? Vysvětlete.

21 SSL/TLS (otázka studijního zaměření – 3 body)

1. Jaké základní bezpečnostní záruky poskytuje protokol SSL/TLS (jinými slovy, jaký je jeho účel) ?
2. Vysvětlete, jak může při komunikaci protokolem SSL/TLS klient ověřit totožnost serveru.
3. Načrtněte algoritmus, kterým se při RSA handshake obě strany spojení dohodnou na sdílené tajné hodnotě (master secret nebo alespoň pre master secret).

22 DNS (otázka studijního zaměření – 3 body)

Uvažujte následující popis DNS zóny:

```
$ORIGIN example.com.
example.com. IN SOA machine.example.com. hostmaster.example.com. ( 2000010101 1d 1h 1w 1h )
example.com. IN NS machine.example.com.
example.com. IN NS machine.elsewhere.com.
example.com. IN MX 10 machine.example.com.
example.com. IN MX 20 another.example.com.
machine IN A 192.0.2.1
another IN A 192.0.2.2
www IN CNAME machine
```

1. K čemu se takový popis DNS zóny používá a na jakých počítačích bude umístěn ?
2. K čemu slouží čísla v závorce na konci SOA záznamu (není nutné přesné vysvětlení každého čísla, stačí obecné vysvětlení jejich účelu) ?
3. K čemu slouží záznamy A, MX a NS v popisu DNS zóny a proč je jich v příkladu několik každého typu ?
4. V záznamu DNS provozu jste si všimli pokusů o nalezení počítače se jménem `1.2.0.192.in-addr.arpa`. Jaké aktivity se záznam týká a jak souvisí s uvedeným popisem DNS zóny ?

23 Spouštění procesů (otázka studijního zaměření – 3 body)

Uvažujte následující program:

```
int main (void) {
    pid_t pid = fork ();
    assert (pid != -1);
    if (pid > 0) {
        int status;
```

```

    waitpid (pid, &status, 0);
    return (status);
} else {
    const char *executable = "/bin/bash";
    execl (executable, executable, NULL);
    assert (0);
}
}
}

```

1. Která z větví příkazu `if` se při spuštění tohoto programu vykoná a proč ?
2. Bude se hodnota proměnné `pid` při opakovaném spouštění tohoto programu také opakovat a proč ?
3. Co by se stalo při spuštění tohoto programu, kdyby řetězec v proměnné `executable` ukazoval na jméno souboru, ve kterém je uložena spustitelná podoba tohoto programu, a proč ?

24 Synchronizace na multiprocесorech (otázka studijního zaměření – 3 body)

Uvažujte program s dvěma vlákny běžící na obvyklém SMP systému. Vlákna jsou plánována tak, že mohou běžet současně. Jedno z nich vykoná funkci `threadOne`, druhé vykoná funkci `threadTwo`, obě vlákna po dokončení těchto funkcí skončí:

```

std::atomic<int> shared;

void threadOne (void) {
    for (int i = 0 ; i < 1000 ; i ++ ) {
        shared ++;
    }
}

void threadTwo (void) {
    for (int i = 0 ; i < 1000 ; i ++ ) {
        shared ++;
    }
}

```

1. Pokud má na začátku běhu obou vláken proměnná `shared` hodnotu 0, je zaručeno, že na konci jejich běhu bude mít právě hodnotu 2000, a proč ?
2. Proměnná `shared` je typu `std::atomic<int>` místo pouhého `int`. Má tato skutečnost vliv na rychlost, kterou obě vlákna vykonají zmíněné funkce, a proč ?
3. Je pro předchozí otázku důležité vědět, zda obě vlákna běží na stejném procesorovém jádru nebo ne, a proč ?
4. Načrtněte, jak byste s použitím zámků docílili stejného chování vláken, pokud by typ proměnné `shared` byl `int` místo `std::atomic<int>` (stejně chování znamená i dostupnost proměnné `shared` mezi iteracemi cyklů).
5. Bude řešení z předchozí otázky rychlejší nebo pomalejší než původní řešení a proč ?

25 Testování funkčnosti (otázka studijního zaměření – 3 body)

V běžném imperativním programovacím jazyce načrtněte implementaci zásobníku s limitem maximální velikosti. Rozhraní by mělo nabízet funkce `push` a `pop`, první akceptuje a druhá vrací referenci na uložený objekt, první vhodně indikuje pokus o překročení maximální velikosti, druhá vhodně indikuje pokus o odebrání objektu z prázdného zásobníku. Načrtněte unit testy, které ověří funkci vaší implementace. Vysvětlete, co se myslí pokrytím testů a určete ho pro vaší implementaci a vaše testy.

Při hodnocení odpovědi nebudou uvažovány drobné syntaktické chyby, ale obecně by použité konstrukce měly odpovídat zvolenému jazyku. Vaše implementace ani vaše testy nemají uvažovat paralelismus.

26 Databáze a web: webová anketa (otázka studijního zaměření – 3 body)

Mějme běžný zpravodajský web, který nabízí články jednak veřejně a jednak podporuje registrované uživatele, kteří se musí autentizovat přihlašovací jménem a heslem. Přihlášení uživatelé vidí u každého článku anketní formulář, kde mohou ohodnotit, jak se jim článek líbí (známkou 1 až 5). Formulář je odeslán klasicky metodou POST na server, kde je zpracován skriptem (např. v PHP) a uložen do databáze. Každý uživatel smí hlasovat nejvýše jednou a články, které již ohodnotil, zobrazí místo ankety průměrné hodnocení článku.

1. Popište alespoň jeden způsob, jakým je možné udržovat uživatelskou relaci webové aplikace (tj. jak si zapamatovat přihlášeného uživatele po jeho autentizaci), když HTTP je ryze bezstavový protokol.
2. Navrhněte fragment HTML pokrývající část stránky s formulářem (drobné syntaktické chyby v HTML budou tolerovány) a vysvětlete, jak jsou tato data přenesena skriptu, který je má zpracovat.
3. Stručně popište, co by mělo být v následné HTTP odpovědi, kterou skript vygeneruje po uložení dat do DB.
4. Stručně popište, jaké změny by bylo potřeba provést na straně klienta a na straně serveru, pokud by měl být formulář odeslán technikou AJAX. Dále popište, jak se tím změní chování aplikace z hlediska uživatele.

27 Databáze a web: SQL (otázka studijního zaměření – 3 body)

Informační systém GoReTex pro automatické vyhodnocování domácích úloh studentů má následující databázové schéma (uvádíme pouze podmnožinu schématu, která je relevantní pro naši úlohu):

`Student(ID, Jmeno, Prijmeni)`

`Uloha(ID, Nazev, CasZadani, CasDeadline, Body)`

`Odevzdani(ID, StudentID, UlohaID, CasOdevzdani, Spravnost)`

Položky ID jsou primární klíče typu integer. Obdobně položky končící na ID jsou cizí klíče do příslušných tabulek (rovněž typu integer). Položky začínající na Cas jsou časová razítka. Body představují maximální počet bodů, které je možné získat za danou úlohu (integer). Spravnost je relativní vyhodnocená správnost řešení v procentech (integer). Zbývající položky jsou řetězce.

Každý student může odevzdat více řešení. Z hlediska bodování se počítá nejúspěšnější řešení odevzdané nejpozději v momentě deadline. Body za odevzdání odpovídají poměrově maximálním bodům za úlohu dle dané správnosti ($\text{Body} * \text{Spravnost} / 100$).

1. Napište SQL dotaz, který vypíše všechny studenty, kteří odevzdali alespoň jedno včasné řešení k úloze s ID = 42, a získané body k této úloze.
2. Upravte předchozí dotaz tak, aby vypsaly všechny studenty. Ti, kteří neodevzdali žádné řešení k úloze s ID = 42 nechť mají v tabulce místo bodů hodnotu NULL.
3. Pokud se data v tabulkách upravují výrazně méně často než se provádí dotazy na body, je dané schéma výkonnostně suboptimální. Přidáme proto tabulku:

`Nejlepsi(StudentID, UlohaID, OdevzdaniID)`

Tato tabulka udržuje pro každého studenta a každou úlohu nejlépe obodované odevzdání. Popište, co je potřeba, aby tato tabulka byla na úrovni databázového systému (tj. bez úprav aplikační vrstvy) udržována aktuální. Není potřeba napsat přesné SQL dotazy, stačí dostatečně přesný slovní popis.

28 Databáze a web: databázové indexy (otázka studijního zaměření – 3 body)

V běžném relačním databázovém systému je schéma obsahující tabulku `Uzivatele` se sloupci `ID`, `Jmeno`, `Prijmeni`, `Vytvoren`, `Aktivni`. ID je 32-bit integer a primární klíč. Jméno a příjmení jsou typu `Varchar(100)` a je na nich vytvořen běžný (dvouatributový) vyhledávací index. Sloupec `Vytvoren` je časové razítko, kdy byl uživatel vložen do systému. Příznak `Aktivni` je booleovská hodnota (tedy integer nabývající hodnot 0 a 1) a tento příznak má také vlastní index.

1. Navrhněte, jak by mohla být data včetně indexů uložena na disku. Pokud použijete známou datovou strukturu (např. B-strom), není třeba popisovat její běžně známé vlastnosti, ale zaměřte se na skutečnou reprezentaci této struktury ve fyzické paměti.

- Předpokládejme, že primární index (ID) používá redundantní B+ strom. Databáze je uložena na filesystému, který má cluster velikosti 4KiB. Navrhněte vhodný parametr m , který určuje stupeň uzlů stromu, a zdůvodněte. Není podstatné exaktní číslo, ale především postup, jak toto číslo určíte.
- Představme si následující SQL dotaz:

```
SELECT CONCAT(Jmeno, ' ', Prijmeni) FROM Uzivatele
WHERE Aktivni = 0 AND Vytvoren > DATEADD(month, -1, GETDATE())
```

Stručně popište (případně schématicky naznačte), jak by mohl vypadat plán provedení tohoto dotazu s přihlédnutím k danému schématu a navržené reprezentaci. Funkce DATEADD a její argumenty nejsou podstatné, v klauzuli ji chápejte jako konstantu.

29 XML (otázka studijního zaměření – 3 body)

Jako kontext pro následující tři specializační otázky uvažujte seznam pracovních inzerátů. Pracovní inzerát umožňuje evidovat název pracovní pozice (text, povinné), požadované zkušenosti (delší text, nepovinné), požadované vzdělání (delší text, povinné), místo (město a země uvedené jako kódy z číselníku měst a zemí, nepovinné s možností uvést více dvojic město a země), minimální a maximální plat v CZK (celá čísla větší než 0, nepovinné) a zaměstnavatel ve struktuře název zaměstnavatele (povinné) a kontaktní email (povinné).

Vytvořte XML schéma popisující XML dokumenty pro zápis pracovních inzerátů uvedené struktury. K XML schématu napište také příklad validního XML dokumentu (s alespoň jedním inzerátem). Vytvořte XSLT skript, který vypíše seznam názvů pracovních inzerátů v podobě setříděného HTML seznamu.

30 REST (otázka studijního zaměření – 3 body)

Stejně jako v předchozí otázce uvažujte seznam pracovních inzerátů. Vysvětlete principy REST na příkladu služby či služeb pro správu pracovních inzerátů reprezentovaných v této struktuře. Uveďte příklad volání pro vytvoření nového inzerátu v seznamu inzerátů, pro získání obsahu jednoho konkrétního inzerátu a pro jeho smazání ze seznamu. Při uvádění příkladů můžete zvolit libovolný vhodný jazyk pro reprezentaci inzerátů.

Dále na příkladu pracovních inzerátů vysvětlete 4 principy Linked Data.

31 Návrhové vzory (otázka studijního zaměření – 3 body)

Stejně jako v předchozí otázce uvažujte seznam pracovních inzerátů. Uvažujte dvě různé třídy (v libovolném objektovém běžně používaném jazyce – C++, C#, Java, ...) reprezentující služby pro správu pracovních inzerátů. První třída, `SluzbaA`, nabízí ve svém rozhraní pro vložení nového inzerátu metodu `vytvorInzerat`, druhá třída, `SluzbaB`, nabízí metodu `zalozNovyInzerat`. Obě mají také různé sady parametrů pro specifikaci nového inzerátu.

Uvažujte třídu `Klient`, jejíž implementace umožňuje vytvořit nový inzerát voláním metody `vytvorInzerat`. Přišel požadavek, aby byla využívána metoda `zalozNovyInzerat`. Navrhněte řešení, které nebude mít vliv na vnitřní kód třídy `Klient` ani třídy `SluzbaB`. Pro řešení zvolte vhodný návrhový vzor. Řešení buď zakreslete v podobě diagramu tříd nebo запиšte v kódu ve zvoleném jazyce.

32 LA1 – Vlastní čísla (3 body)

- Buď

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 3 & 0 & 1 \end{pmatrix}.$$

Rozhodněte, zda $\lambda = 7$ je vlastním číslem matice A . Dále rozhodněte, zda $x = (1, 2, 1)^T$ je vlastním vektorem matice A .

- Buď V množina všech reálných matic 3×3 , pro které je 0 vlastní číslo a přísluší mu vlastní vektor $y = (1, 2, 3)^T$. Ukažte, že V je vektorový prostor a určete jeho dimenzi.

33 LA2 – Pozitivní definitnost (3 body)

Definujte pojem pozitivní definitnosti matice a formulujte alespoň jedno ekvivalentní kritérium pro testování pozitivní definitnosti. Pomocí tohoto kritéria rozhodněte, zda je následující matice pozitivně definitní

$$B = \begin{pmatrix} 4 & -2 & 0 & 2 \\ -2 & 2 & 2 & -1 \\ 0 & 2 & 5 & -2 \\ 2 & -1 & -2 & 6 \end{pmatrix}$$

34 MA1 – Limita posloupnosti (3 body)

Definujte vlastní limitu posloupnosti.

Nechť $(a_n)_{n=1}^{\infty}$ a $(b_n)_{n=1}^{\infty}$ jsou posloupnosti reálných čísel. Rozhodněte zda platí, že pokud $\lim_{n \rightarrow \infty} (a_n - b_n) = 0$, pak $\lim_{n \rightarrow \infty} \frac{a_n}{b_n} = 1$. Své rozhodnutí zdůvodněte.

35 MA2 – Metrický prostor (3 body)

Dokažte, že (M, d) je metrický prostor pokud M je neprázdná množina, f je funkce z M do kladných reálných čísel a d je definována jako

$$d(x, y) = \begin{cases} 0 & x = y \\ \max(f(x), f(y)) & x \neq y. \end{cases}$$

36 MA3 – Tečná rovina (3 body)

Najděte bod, kde je tečná rovina funkce $f(x, y) = x^2 + 2xy + 2y^2 - 6x - 8y$ vodorovná. Jaka je rovnice této tečné roviny?

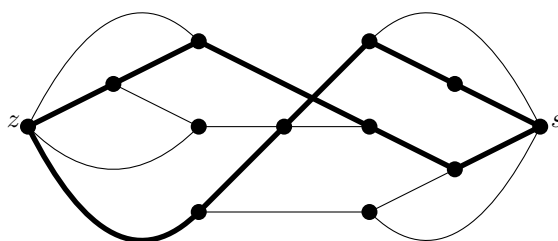
37 Nezávislé jevy (3 body)

1. Co znamená, že dva náhodné jevy jsou nezávislé? Definujte.
2. Nechť $n \geq 2$ je přirozené číslo. Zkonstruujeme náhodný graf G s množinou vrcholů $\{1, 2, \dots, n\}$ tak, že pro každou dvojici vrcholů i a j takových, že $1 \leq i < j \leq n$, si hodíme férovou mincí (na níž padne hlava s pravděpodobností přesně $1/2$) a padne-li na ní hlava, přidáme do G hranu ij . Nechť J_i je jev, že vrchol i má stupeň přesně 1. Rozhodněte, pro která n jsou jevy J_1 a J_2 nezávislé a odpověď zdůvodněte.

38 Grafy (3 body)

Z grafu G na obrázku vytvoříme síť tak, že všechny hrany orientujeme zleva doprava a přidělíme jim kapacitu 1.

Tok mezi z a s je vyznačen silnými hranami a využívá plnou kapacitu vyznačených hran. Kapacity ostatních hran nejsou využity vůbec.



Rozhodněte, zdali jde o maximální tok. Pokud vyznačený tok není maximální, nějaký maximální tok nalezněte a zdůvodněte jeho maximalitu.

Jaké jsou důsledky hodnoty maximálního toku pro hranovou souvislost grafu ?

39 Kombinatorika (3 body)

Nechť pro $n \in \mathbb{N}$ značí $f(n)$ počet 15-prvkových podmnožin n -prvkové množiny, a necht' $g(n)$ značí počet zobrazení z n -prvkové množiny do tříprvkové.

Vyjádřete $f(n)$ a $g(n)$.

Rozhodněte, která z následujících možností platí:

- $\exists n_0 : \forall n > n_0 : f(n) > g(n)$
- $\exists n_0 : \forall n > n_0 : g(n) > f(n)$
- žádné n_0 vyhovující předchozím dvěma podmínkám neexistuje.

40 Logika (3 body)

Mějme následující formule φ_1, φ_2 jazyka $L = \langle 0, |, f, -, < \rangle$ s rovnostmi, kde 0 je konstantní symbol, $|, f$ jsou unární funkční symboly, $-$ je binární funkční a $<$ je binární relační symbol

$$\varphi_1 : (\forall \varepsilon)(\forall u)(0 < \varepsilon \rightarrow (\exists \delta)(0 < \delta \wedge (\forall x)(|x - u| < \delta \rightarrow |f(x) - f(u)| < \varepsilon))),$$

$$\varphi_2 : (\exists u)(\exists \varepsilon)(0 < \varepsilon \wedge (\forall \delta)(0 < \delta \rightarrow (\exists x)(|x - u| < \delta \wedge \neg(|f(x) - f(u)| < \varepsilon)))).$$

1. Uveďte definice pro predikátovou logiku, kdy formule φ platí ve struktuře \mathcal{A} , kdy φ (logicky) platí, a kdy je φ nezávislá.
2. Platí formule φ_1, φ_2 ve struktuře $\mathcal{A} = \langle \mathbb{R}, 0, |, f, -, < \rangle$ jazyka L , kde 0, $|, -, <$ má svůj obvyklý význam na \mathbb{R} a $f(0) = 0, f(r) = |r|/r$ pro $r \neq 0$? Uveďte zdůvodnění.
3. Je formule $\varphi_1 \leftrightarrow \neg\varphi_2$ nezávislá ? Uveďte zdůvodnění.