

# Korespondenční Seminář z Programování

## ZAČÁTEČNICKÁ KATEGORIE



28. ročník

KSP-Z

Květen 2015

Právě se díváte do letáku první série 28. ročníku KSP-Z, neboli Korespondenčního Semináře z Programování, Začátečnické kategorie. Jedná se o korespondenční soutěž spočívající v řešení jednodušších programátorských úloh, které vychází během školního roku. Zapojit se může každý středoškolák i základoškolák, ty úspěšnější z vás pak na jaře pozveme na týdenní soustředění, na kterém se toho spoustu dozvíte a zároveň si užijete hromadu neopakovatelné zábavy. Pokud budete mít jakoukoliv otázku, neváhejte se zeptat. Kontaktní adresy najdete v tiráži na konci letáku. Přejeme hodně štěstí!

**Termín série: 23. listopadu 2015 v 8:00 SEČ**

**Obsah série: 4 praktické úlohy** (značené ) – K těmto úlohám je nutné napsat program (v libovolném vhodném jazyce), stáhnout si z našeho webu vstupní data a odevzdat odpovídající výstup.  
**2 teoretické úlohy** (značené ) – U těchto úloh nás zajímá hlavně slovní popis řešení, ve kterém byste měli zdůvodnit jeho funkčnost a ideálně nás i přesvědčit o jeho efektivitě.



**Odevzdávání:** Přes web na adrese <https://ksp.mff.cuni.cz/z/odevzdej.cgi>.

### Zadání první série začátečnické kategorie 28. ročníku KSP

#### 28-Z1-1 Kevinův leták 8 bodů

Kevin otevřel obálku (kterou našel ve schránce) a vyzval z ní všechny papíry. Jeden ho obzvláště zaujal (ten od KSP). Ihned si všiml, že (na začátku) obsahuje nějak mnoho závorek. Tak by ho zajímalo, jestli tam (autoři zadání) neudělali nějakou chybu. Pomůžete mu (s ověřením)?

Pro danou posloupnost symbolů (  $($  ,  $)$  ), tedy otevíracích a zavíracích závorek, najděte od začátku co nejdelší úsek, který je platným uzavřováním – tedy každá závorka má svoji do páru.

*Tvar vstupu:* Na prvním řádku dostanete číslo  $N$ . Na dalších  $N$  řádcích budou jednotlivé testovací případy.

*Tvar výstupu:* Pro každý testovací případ vypište na řádek délku nalezeného uzavřování.

Slibujeme, že  $N$  bude nejvýše 1000, a každý řádek bude mít nejvýše  $10^5$  symbolů.

*Ukázkový vstup:*

```
3
( ( ) ) ( ) ( )
(( ( ) ) ( ) )
( ( ) ) ( ) ( ) ( )
```

8

12

6

Uzavírací závorka na deváté pozici nemá svůj protějšek. Druhý řádek je celý platně uzavřování.

#### 28-Z1-2 Sárina hra 10 bodů

Znáte to, když jedete autobusem, cesta občas ubíhá strašlivě pomalu. Potom vymyslíte možné i nemožné, abyste se zabavili. Při jedné takové cestě Sáru napadla hříčka.

Pořídila si seznam  $N$  zastávek po cestě. Seznam si očíslovala tak, že zastávka, na které nastoupila, dostala číslo jedna, a ta, na které vystoupila, číslo  $N$ . Pokaždé, když autobus přijel do zastávky  $k$ , si Sára zakroužkovala každou  $k$ -tou zastávku v seznamu – tedy zastávky číslo  $k, 2k, 3k, \dots$ . Pokud už nějaká zastávka zakroužkována byla, zakroužkování zase vygumovala.

Uznejte sami, není to úplně zábavná hra. Také to Sáru před  $M$ -tou zastávkou přestalo bavit. Co ale zastávka, na které vystupujete? Je zakroužkována?

*Tvar vstupu a výstupu:* Na prvním řádku je číslo  $J$  udávající počet úloh, které budete řešit, těch bude nejvýše 1000. Na

dalších  $J$  řádcích najdete mezerou oddělená čísla  $N$  a  $M$ , kde  $1 < M \leq N < 10^9$ . Pro každý řádek si zahrajte Sárinu hru, a vypište na výstup ANO, pokud byla  $N$ -tá zastávka na konci hry zakroužkována, jinak NE.

*Ukázkový vstup:*

```
3
9 4
6 4
7 7
```

*Ukázkový výstup:*

ANO

NE

NE

Devátou zastávku Sára zakroužkovala po projetí třetí, zato šestou po třetí zastávce vygumovala. Vě třetím řádku ji kroužkování přestalo bavit těsně před tím, než by sedmíčku zakroužkovala poprvé.



#### 28-Z1-3 Petrovy stromy 10 bodů

Petr raději při cestě autobusem kouká z okna a počítá. Co počítá? Srnky, posezy, stromy... Počkat, stromy?

Podél cesty je vysázeno několik druhů stromů. Následují po sobě pravidelně, jako neustále se opakující vzorec. Dub, topol, dub, topol, dva duby a buk. A tak pořád dokola. Potom, co tohle chvíli sleduje, se mu začínají dělat mžítka před očima.

Petr by rád věděl, za jak dlouho uvidí  $K$ -tý strom daného druhu. Protože ale nezná rychlost autobusu, stačí mu vědět, kolikátý strom v řadě to bude. Vypočítáte to pro něj?

*Tvar vstupu:* Na prvním řádku budou čísla  $B$ ,  $J$  a  $K$  oddělená mezerou.  $B$  je počet stromů v bloku, který se opakuje, a  $J$  je číslo druhu stromu, jehož  $K$ -tý výskyt Petra zajímá. Máte slíbeno, že  $B$  bude nejvýše  $10^4$ ,  $J$  nejvýše 1000 a  $K$  nebude větší než  $10^8$ .

Na dalším řádku bude  $B$  čísel  $1 \dots 1000$  označujících druhy stromů, které se podél cesty opakují.

*Tvar výstupu:* Vypište jediné číslo  $L$  takové, že  $K$ -tý výskyt druhu  $J$  je  $L$ -tý v řadě všech stromů. Předpokládejte při-

tom, že řada stromů je dostatečně dlouhá, a daný strom se podél cesty vyskytuje.

*Ukázkový vstup:*

```
7 2 5
1 2 1 2 1 1 3
```

*Ukázkový výstup:*

16

Příklad odpovídá ukázce z textu. Pátý topol (druh číslo 2), který Petr uvidí, bude šestnáctý v řadě.

#### 28-Z1-4 Zuzčina zvědavost 12 bodů

Malá Zuzka nedávno oslavila šesté narozeniny, takže v září nastupuje do první třídy. Už byla u zápisu, stejně jako všichni z jejího ročníku. Každý věděl, s kým bude chodit do třídy, a s každým svým spolužákem se při zápisu seznámil. S nikým jiným se nikdo neseznamoval.

Zuzka s Kevinem chvíli pozorovali seznamování ostatních. Potom ale Zuzka položila zvědavý dotaz: „Kolik vlastně bude prvních tříd?“ Ještě, že si Kevin zapsal několik dvojic, které viděl se seznamovat. Pomozte mu ze zapsaných dat vykoukat odpověď na Zuzčinu otázku.

*Tvar vstupu:* První řádek obsahuje čísla  $N$  a  $M$ , kde  $N$  je počet různých dětí, které Kevin viděl, a  $M$  je počet seznamujících se dvojic, které si zapsal. Víte, že  $1 < N < 1000$  a žádná dvojice není v seznamu dvakrát. Následuje seznam  $M$  dvojic čísel oddělených mezerou. Každé dítě je pro jednoduchosť označeno číslem  $1 \dots N$ .

*Tvar výstupu:* Vypište jedno číslo odpovídající největšímu možnému počtu tříd, které škola podle Kevinova pozorování může otevřít. Každá třída má alespoň jednoho žáka.

*Ukázkový vstup:*

```
5 3
1 2
2 3
4 5
```

*Ukázkový výstup:*

2

Jednu třídu tvoří žáci 1, 2 a 3, druhou pak 4 a 5.

#### 28-Z1-5 Dvě fronty na oběd 12 bodů

Když už měla Zuzka dost pozorování, šla s ostatními na oběd. V jídelně mají takové zvláštní pravidlo – fronta žáků čekajících na oběd musí být neustále seřazena podle výšky, aby kuchařky viděly všem najednou do obličejů.

Cestou na oběd se tedy skupina, ve které Zuzka šla, správně seřadila. Jenže možné cesty na oběd jsou dvě – u jídelny se potkali s jinou skupinkou. Ta byla přibližně stejně velká, a také už byla správně seřazena. Jak se mají obě skupinky spojit do výsledné fronty tak, aby byli žáci seřazeni?

Protože jsou všichni hladoví, zkuste vymyslet postup, který vyžaduje co nejméně porovnání výšek dvou žáků – to je totiž to jediné, co s výškami můžete dělat.

<sup>1</sup> <http://ksp.mff.cuni.cz/viz/kucharky/zakladni-algoritmy>

<sup>2</sup> <http://ksp.mff.cuni.cz/viz/kucharky/slozitost>

Formálněji, máme dvě seřazená pole a chceme z nich sestojit jedno obsahující všechny položky z obou, také správně seřazené.

Zajímá nás *asymptotická složitost* vašeho řešení v nehorším případě. To znamená řádový počet operací pro nehorší možný případ, vyjádřený pomocí proměnných. Například si můžete označit proměnnými  $A$  a  $B$  velikosti skupin a říct, že k seřazení potřebujete řádově  $(A+B)^2$  operací. Podrobněji o složitosti se dočtete v naší kuchařce, kterou vřele doporučujeme přečíst od začátku – nejprve základní algoritmy,<sup>1</sup> poté složitost.<sup>2</sup>

Aby úloha nebyla zbytečně těžká, mají žáci k řešení dostatek prostoru. Tedy například můžete ze dvou front stavět jednu výslednou někde úplně jinde.

#### 28-Z1-6 Osm čipů 14 bodů

Zatímco Zuzka se mohla jít najíst, Kevin pomáhal s přípravou na prvníčky. Jedním z úkolů, který řešil, bylo přidělení čipů na přístup do školy. Takový čip má zvolené unikátní číslo a jeho přečtení čtečkou otevře dveře.

Kevin si vymyslel speciální způsob přidělování čísel. Na začátku si zvolí tři libovolná celá čísla  $a$ ,  $b$  a  $c$ . Následně vždy spočítá číslo nového čipu  $x_i$  z posledních třech čísel vzorcem  $x_i = ax_{i-3} + bx_{i-2} + cx_{i-1}$ . Čísla prvních tří čipů si prostě vymyslí libovolná, podle toho, jakou má náladu.

Pak si ale vzpomněl, že čip má v sobě omezenou paměť, takže nepojme libovolně velké číslo. Místo původního čísla tedy zapíše do čipu jeho zbytek po vydělení  $N$ , kde  $N$  je nějaké rozumné velké číslo.

Pak si ale uvědomil, že tento postup nezaručuje, že každý čip bude mít čísla různá od ostatních. Už kvůli tomu, že může existovat nejvýše  $N$  čipů s různými čísly, takže se jednou musí čísla začít opakovat.

Ze zvědavosti si Kevin vymyslel číslo  $K$ , které je řádově větší než  $N$ , a ptá se vás – jaké číslo by měl  $K$ -tý čip, který Kevin takto vytvoří? Vymyslete způsob, jak spočítat  $x_K$  co nejrychleji.

Předpokládejte, že máte počítač s dostatkem paměti. Naopak nepředpokládejte, že  $K$  je rozumně malé. Kevinův počítač umí s tak velkými čísly provádět základní operace, ale například nestihne do  $K$  napočítat po jedné.

*Příklad:* Zvolme si parametry:  $a = 0$ ,  $b = 1$ ,  $c = 1$ ,  $N = 3$  a  $K = 52$ . Prvním čipům dejme čísla 1, 1 a 2. Kevin bude postupně přidělovat čísla takto:

1, 1, 2, 0, 2, 2, 1, 0, 1, 1, 2, 0, 2, 2, 1, 0, 1, 1, 2, ...

Padesátý druhý čip dostane číslo 0.



KSP pro vás připravují studenti Matematicko-fyzikální fakulty Univerzity Karlovy v Praze.

**Webové stránky:**

<https://ksp.mff.cuni.cz/>

**E-mail:**

[ksp@mff.cuni.cz](mailto:ksp@mff.cuni.cz)

**Diskusní fórum:**

<https://ksp.mff.cuni.cz/forum/>

Chcete-li s námi komunikovat bezpečně, můžete si ověřit náš HTTPS certifikát – jeho SHA1 fingerprint je: 0E:D9:B6:E5:6F:BF:05:1D:99:66:EB:E9:29:E4:58:AB:5F:99:6D:FD:A3.



# Korespondenční Seminář z Programování

## ZAČÁTEČNICKÁ KATEGORIE



28. ročník

KSP-Z

Květen 2015

Právě se díváte do letáku první série 28. ročníku KSP-Z, neboli Korespondenčního Semináře z Programování, Začátečnické kategorie. Jedná se o korespondenční soutěž spočívající v řešení jednodušších programátorských úloh, které vychází během školního roku. Zapojit se může každý středoškolák i základoškolák, ty úspěšnější z vás pak na jaře pozveme na týdenní soustředění, na kterém se toho spoustu dozvíte a zároveň si užijete hromadu neopakovatelné zábavy. Pokud budete mít jakoukoliv otázku, neváhejte se zeptat. Kontaktní adresy najdete v tiráži na konci letáku. Přejeme hodně štěstí!

**Termín série: 23. listopadu 2015 v 8:00 SEČ**

**Obsah série: 4 praktické úlohy** (značené ) – K těmto úlohám je nutné napsat program (v libovolném vhodném jazyce), stáhnout si z našeho webu vstupní data a odevzdat odpovídající výstup.  
**2 teoretické úlohy** (značené ) – U těchto úloh nás zajímá hlavně slovní popis řešení, ve kterém byste měli zdůvodnit jeho funkčnost a ideálně nás i přesvědčit o jeho efektivitě.

**Odevzdávání:** Přes web na adrese <https://ksp.mff.cuni.cz/z/odevzdej.cgi>.



### Zadání první série začátečnické kategorie 28. ročníku KSP

#### 28-Z1-1 Kevinův leták 8 bodů

Kevin otevřel obálku (kterou našel ve schránce) a vyzval z ní všechny papíry. Jeden ho obzvláště zaujal (ten od KSP). Ihned si všiml, že (na začátku) obsahuje nějak mnoho závorek. Tak by ho zajímalo, jestli tam (autoři zadání) neudělali nějakou chybu. Pomůžete mu (s ověřením)?

Pro danou posloupnost symbolů (  $($  ,  $)$  ), tedy otevíracích a zavíracích závorek, najděte od začátku co nejdelší úsek, který je platným uzávorkováním – tedy každá závorka má svoji do páru.

*Tvar vstupu:* Na prvním řádku dostanete číslo  $N$ . Na dalších  $N$  řádcích budou jednotlivé testovací případy.

*Tvar výstupu:* Pro každý testovací případ vypište na řádek délku nalezeného uzávorkování.

Slibujeme, že  $N$  bude nejvýše 1000, a každý řádek bude mít nejvýše  $10^5$  symbolů.

*Ukázkový vstup:*

```
3
( ) ( ) ( ) ( )
(( ) ( ) ( ) ( )
( ( ) ( ) ( ) ( ) ( )
```

8

12

6

Uzavírací závorka na deváté pozici nemá svůj protějšek. Druhý řádek je celý platný uzávorkování.

#### 28-Z1-2 Sárina hra 10 bodů

Znáte to, když jedete autobusem, cesta občas ubíhá strašlivě pomalu. Potom vymyslíte možné i nemožné, abyste se zabavili. Při jedné takové cestě Sáru napadla hříčka.

Pořídila si seznam  $N$  zastávek po cestě. Seznam si očíslovala tak, že zastávka, na které nastoupila, dostala číslo jedna, a ta, na které vystoupila, číslo  $N$ . Pokaždé, když autobus přijel do zastávky  $k$ , si Sára zakroužkovala každou  $k$ -tou zastávku v seznamu – tedy zastávky číslo  $k, 2k, 3k, \dots$ . Pokud už nějaká zastávka zakroužkována byla, zakroužkování zase vygumovala.

Uznejte sami, není to úplně zábavná hra. Také to Sáru před  $M$ -tou zastávkou přestalo bavit. Co ale zastávka, na které vystupujete? Je zakroužkována?

*Tvar vstupu a výstupu:* Na prvním řádku je číslo  $J$  udávající počet úloh, které budete řešit, těch bude nejvýše 1000. Na

dalších  $J$  řádcích najdete mezerou oddělená čísla  $N$  a  $M$ , kde  $1 < M \leq N < 10^9$ . Pro každý řádek si zahrajte Sárinu hru, a vypište na výstup ANO, pokud byla  $N$ -tá zastávka na konci hry zakroužkována, jinak NE.

*Ukázkový vstup:*

```
3
9 4
6 4
7 7
```

*Ukázkový výstup:*

ANO

NE

NE

Devátou zastávku Sára zakroužkovala po projetí třetí, zato šestou po třetí zastávce vygumovala. Vě třetím řádku ji kroužkování přestalo bavit těsně před tím, než by sedmíčku zakroužkovala poprvé.



#### 28-Z1-3 Petrovy stromy 10 bodů

Petr raději při cestě autobusem kouká z okna a počítá. Co počítá? Srnky, posezy, stromy... Počkat, stromy?

Podél cesty je vysázeno několik druhů stromů. Následují po sobě pravidelně, jako neustále se opakující vzorec. Dub, topol, dub, topol, dva duby a buk. A tak pořád dokola. Potom, co tohle chvíli sleduje, se mu začínají dělat mžítka před očima.

Petr by rád věděl, za jak dlouho uvidí  $K$ -tý strom daného druhu. Protože ale nezná rychlost autobusu, stačí mu vědět, kolikátý strom v řadě to bude. Vypočítáte to pro něj?

*Tvar vstupu:* Na prvním řádku budou čísla  $B$ ,  $J$  a  $K$  oddělená mezerou.  $B$  je počet stromů v bloku, který se opakuje, a  $J$  je číslo druhu stromu, jehož  $K$ -tý výskyt Petra zajímá. Máte slíbeno, že  $B$  bude nejvýše  $10^4$ ,  $J$  nejvýše 1000 a  $K$  nebude větší než  $10^8$ .

Na dalším řádku bude  $B$  čísel  $1 \dots 1000$  označujících druhy stromů, které se podél cesty opakují.

*Tvar výstupu:* Vypište jediné číslo  $L$  takové, že  $K$ -tý výskyt druhu  $J$  je  $L$ -tý v řadě všech stromů. Předpokládejte při-

tom, že řada stromů je dostatečně dlouhá, a daný strom se podél cesty vyskytuje.

*Ukázkový vstup:*

```
7 2 5
1 2 1 2 1 1 3
```

*Ukázkový výstup:*

16

Příklad odpovídá ukázce z textu. Pátý topol (druh číslo 2), který Petr uvidí, bude šestnáctý v řadě.

#### 28-Z1-4 Zuzčina zvědavost 12 bodů

Malá Zuzka nedávno oslavila šesté narozeniny, takže v září nastupuje do první třídy. Už byla u zápisu, stejně jako všichni z jejího ročníku. Každý věděl, s kým bude chodit do třídy, a s každým svým spolužákem se při zápisu seznámil. S nikým jiným se nikdo neseznamoval.

Zuzka s Kevinem chvíli pozorovali seznamování ostatních. Potom ale Zuzka položila zvědavý dotaz: „Kolik vlastně bude prvních tříd?“ Ještě, že si Kevin zapsal několik dvojic, které viděl se seznamovat. Pomozte mu ze zapsaných dat vykoukat odpověď na Zuzčinu otázku.

*Tvar vstupu:* První řádek obsahuje čísla  $N$  a  $M$ , kde  $N$  je počet různých dětí, které Kevin viděl, a  $M$  je počet seznamujících se dvojic, které si zapsal. Víte, že  $1 < N < 1000$  a žádná dvojice není v seznamu dvakrát. Následuje seznam  $M$  dvojic čísel oddělených mezerou. Každé dítě je pro jednoduchosť označeno číslem  $1 \dots N$ .

*Tvar výstupu:* Vypište jedno číslo odpovídající největšímu možnému počtu tříd, které škola podle Kevinova pozorování může otevřít. Každá třída má alespoň jednoho žáka.

*Ukázkový vstup:*

```
5 3
1 2
2 3
4 5
```

*Ukázkový výstup:*

2

Jednu třídu tvoří žáci 1, 2 a 3, druhou pak 4 a 5.

#### 28-Z1-5 Dvě fronty na oběd 12 bodů

Když už měla Zuzka dost pozorování, šla s ostatními na oběd. V jídelně mají takové zvláštní pravidlo – fronta žáků čekajících na oběd musí být neustále seřazena podle výšky, aby kuchařky viděly všem najednou do obličejů.

Cestou na oběd se tedy skupina, ve které Zuzka šla, správně seřadila. Jenže možné cesty na oběd jsou dvě – u jídelny se potkali s jinou skupinkou. Ta byla přibližně stejně velká, a také už byla správně seřazena. Jak se mají obě skupinky spojit do výsledné fronty tak, aby byli žáci seřazeni?

Protože jsou všichni hladoví, zkuste vymyslet postup, který vyžaduje co nejméně porovnání výšek dvou žáků – to je totiž to jediné, co s výškami můžete dělat.

<sup>1</sup> <http://ksp.mff.cuni.cz/viz/kucharky/zakladni-algoritmy>

<sup>2</sup> <http://ksp.mff.cuni.cz/viz/kucharky/slozitost>

Formálněji, máme dvě seřazená pole a chceme z nich sestojit jedno obsahující všechny položky z obou, také správně seřazené.

Zajímá nás *asymptotická složitost* vašeho řešení v nehorším případě. To znamená řádový počet operací pro nehorší možný případ, vyjádřený pomocí proměnných. Například si můžete označit proměnnými  $A$  a  $B$  velikosti skupin a říct, že k seřazení potřebujete řádově  $(A+B)^2$  operací. Podrobněji o složitosti se dočtete v naší kuchařce, kterou vřele doporučujeme přečíst od začátku – nejprve základní algoritmy,<sup>1</sup> poté složitost.<sup>2</sup>

Aby úloha nebyla zbytečně těžká, mají žáci k řešení dostatek prostoru. Tedy například můžete ze dvou front stavět jednu výslednou někde úplně jinde.

#### 28-Z1-6 Osm čipů 14 bodů

Zatímco Zuzka se mohla jít najíst, Kevin pomáhal s přípravou na prvníčky. Jedním z úkolů, který řešil, bylo přidělení čipů na přístup do školy. Takový čip má zvolené unikátní číslo a jeho přečtení čtečkou otevře dveře.

Kevin si vymyslel speciální způsob přidělování čísel. Na začátku si zvolí tři libovolná celá čísla  $a$ ,  $b$  a  $c$ . Následně vždy spočítá číslo nového čipu  $x_i$  z posledních třech čísel vzorcem  $x_i = ax_{i-3} + bx_{i-2} + cx_{i-1}$ . Čísla prvních tří čipů si prostě vymyslí libovolná, podle toho, jakou má náladu.

Pak si ale vzpomněl, že čip má v sobě omezenou paměť, takže nepojme libovolně velké číslo. Místo původního čísla tedy zapíše do čipu jeho zbytek po vydělení  $N$ , kde  $N$  je nějaké rozumné velké číslo.

Pak si ale uvědomil, že tento postup nezaručuje, že každý čip bude mít čísla různá od ostatních. Už kvůli tomu, že může existovat nejvýše  $N$  čipů s různými čísly, takže se jednou musí čísla začít opakovat.

Ze zvědavosti si Kevin vymyslel číslo  $K$ , které je řádově větší než  $N$ , a ptá se vás – jaké číslo by měl  $K$ -tý čip, který Kevin takto vytvoří? Vymyslete způsob, jak spočítat  $x_K$  co nejrychleji.

Předpokládejte, že máte počítač s dostatkem paměti. Naopak nepředpokládejte, že  $K$  je rozumně malé. Kevinův počítač umí s tak velkými čísly provádět základní operace, ale například nestihne do čísla napočítat po jedné.

*Příklad:* Zvolme si parametry:  $a = 0$ ,  $b = 1$ ,  $c = 1$ ,  $N = 3$  a  $K = 52$ . Prvním čipům dejme čísla 1, 1 a 2. Kevin bude postupně přidělovat čísla takto:

1, 1, 2, 0, 2, 2, 1, 0, 1, 1, 2, 0, 2, 2, 1, 0, 1, 1, 2, ...

Padesátý druhý čip dostane číslo 0.



KSP pro vás připravují studenti Matematicko-fyzikální fakulty Univerzity Karlovy v Praze.

**Webové stránky:**

<https://ksp.mff.cuni.cz/>

**E-mail:**

[ksp@mff.cuni.cz](mailto:ksp@mff.cuni.cz)

**Diskusní fórum:**

<https://ksp.mff.cuni.cz/forum/>

Chcete-li s námi komunikovat bezpečně, můžete si ověřit náš HTTPS certifikát – jeho SHA1 fingerprint je: 0E:D9:B6:E5:6F:BF:05:1D:99:66:EB:E9:29:E4:58:AB:5F:99:6D:FD:A3.

