

Algoritmy a datové struktury I

1. cvičení

čtvrtek 25. 2. 2016 9:00

Příklady

Úvodní příklad

Máte zadánu posloupnost celých čísel x_1, x_2, \dots, x_n . Jak v ní najdeme souvislý úsek s největším součtem? Jak rychlý je váš algoritmus?

Složitost:

- Jak měříme (časovou) složitost?
- Výpočetní modely – Random access machine (RAM), Pointer machine, Turingův stroj.
- Co znamená $\mathcal{O}(n)$, $\Omega(n)$, $o(n)$, $\omega(n)$ a $\Theta(n)$?

Jak tvrdé je vajíčko?

Máme n -patrový dům a víme, že hodíme-li vajíčko z alespoň k -tého patra, rozbije se. Známe n a chceme zjistit k . Jak to udělat na co nejméně pokusů, máme-li k dispozici jediné vajíčko? Dvě vajíčka? Tři? Libovolně mnoho vajíček? (Pokus = hození vajíčka z nějakého patra.)

Fibonacciho čísla

Navrhněte algoritmus, který co nejrychleji spočte n -té Fibonacciho číslo. (Pro účely tohoto cvičení trvají aritmetické operace jeden krok bez ohledu na velikost čísel s nimiž pracují.)

Nejlepší a nejhorší případ

Jaké znáte třídící algoritmy? Insert, Select, Bubble – zkuste rozmyslet, jaká je jejich složitost v nejlepším a nejhorším případě a jak takové případy vypadají.

Největší díra v matici

Mějme matici velikost $n \times m$. Díra v matici je souvislá podmatice samých nul. Jak co nejrychleji najít tu největší díru v zadané matici?

Organizace a podmínky zápočtu

- Pište mi na husek+ads@iuuk.mff.cuni.cz.
- Web cvičení je na <http://iuuk.mff.cuni.cz/~husek/vyuka/ads1516.html>
- Na zápočet potřebujete nasbírat **100 bodů**, ty můžete získat za:
 - **řešení** domácích úkolů a
 - **zápočtovou práci**.
- Úkoly přijímám **pouze** elektronicky, buď jako prostý text (tzn. přímo v mailu), nebo jako PDF (ideálně vyrobené (La)TeXem, ale to není povinné, jen silně doporučené). Nechci formáty `.doc`, `.docx`, `.odt` atp.
- Úkolů bude vypsáno za cca 170 bodů, nebudou zadávány každý týden, spíš každých 14 dní. Po následující dva týdny od zadání za ně bude možno získat plný počet bodů, poté až do konce semestru můžete úlohy odevzdávat za polovinu bodů.
- Zápočtová práce může být praktická (implementace algoritmu) či teoretická (popis algoritmu včetně pseudokódu, důkaz správnosti, analýza časové složitosti) a bude za ni možno získat podle náročnosti 20–40 bodů. Pár návrhů bude dostupných na webu, ale nejlepší je, když přijdete sami s něčím, co vás zajímá, rád vám vyjdu vstříc.