PAL: 2. cvičení

Tomáš Sieger

1. 10. 2020

Organizace

- materiály, úlohy k řešení
  - samostatné řešení úloh (i dopředu)
  - konzultace
- aktivní účast
  - odevzdávání řešených úkolů **do neděle**
  - aktivita na cvičeních
- práce ve skupinách
  - volba skupinek?
  - “podmístnosti” v BBB
  - sdílené poznámky
  - dotazy v chatu v hlavní místnosti

Opakování z minula

Př. 2 . Porovnání funkcí

$\log(n!)$
$(\sqrt{2})^{\log(n)}$
$2^{\log(\log(n))}$
$4^{\log(n)}$
$\sqrt{(\log(n))}$
$n\log(n^2)$
$n\log(n)$
$(\log(n))^2$

Př. 2 . Porovnání funkcí

$\log(n!)$	$O(n\log(n))$
$(\sqrt{2})^{\log(n)}$	$\sqrt{n}$
$2^{\log(\log(n))}$	$\log(n)$
$4^{\log(n)}$	n^2
$\sqrt{(\log(n))}$	$\sqrt{\log(n)}$
$n\log(n^2)$	$2n\log(n)$
$n\log(n)$	$n\log(n)$
$(\log(n))^2$	$(\log(n))^2$

Př. 2 . Porovnání funkcí

		$n = 2$
$\log(n!)$	$O(n\log(n))$	1
$(\sqrt{2})^{\log(n)}$	$\sqrt{n}$	1.4
$2^{\log(\log(n))}$	$\log(n)$	1
$4^{\log(n)}$	n^2	4
$\sqrt{(\log(n))}$	$\sqrt{\log(n)}$	1.41
$n\log(n^2)$	$2n\log(n)$	4
$n\log(n)$	$n\log(n)$	2
$(\log(n))^2$	$(\log(n))^2$	1

Př. 2 . Porovnání funkcí

		$n=2$	$n=2^{10}$
$\log(n!)$	$O(n\log(n))$	1	8769
$(\sqrt{2})^{\log(n)}$	$\sqrt{n}$	1.4	32
$2^{\log(\log(n))}$	$\log(n)$	1	10
$4^{\log(n)}$	n^2	4	1048576
$\sqrt{(\log(n))}$	$\sqrt{\log(n)}$	1.41	3.16
$n\log(n^2)$	$2n\log(n)$	4	20480
$n\log(n)$	$n\log(n)$	2	10240
$(\log(n))^2$	$(\log(n))^2$	1	100

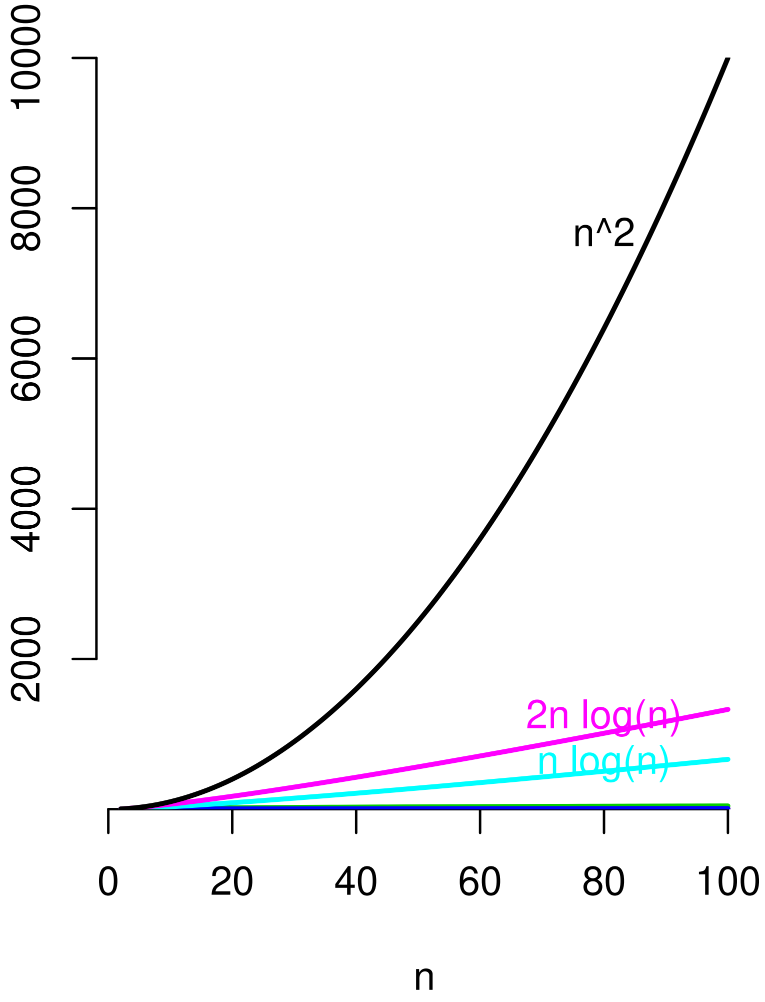
Př. 2 . Porovnání funkcí

		$n=2$	$n=2^{10}$	$n=2^{20}$
$\log(n!)$	$O(n\log(n))$	1	8769	19254028
$(\sqrt{2})^{\log(n)}$	$\sqrt{n}$	1.4	32	1024
$2^{\log(\log(n))}$	$\log(n)$	1	10	20
$4^{\log(n)}$	n^2	4	1048576	1.1010^{12}
$\sqrt{(\log(n))}$	$\sqrt{\log(n)}$	1.41	3.16	4.47
$n\log(n^2)$	$2n\log(n)$	4	20480	41943040
$n\log(n)$	$n\log(n)$	2	10240	20971520
$(\log(n))^2$	$(\log(n))^2$	1	100	400

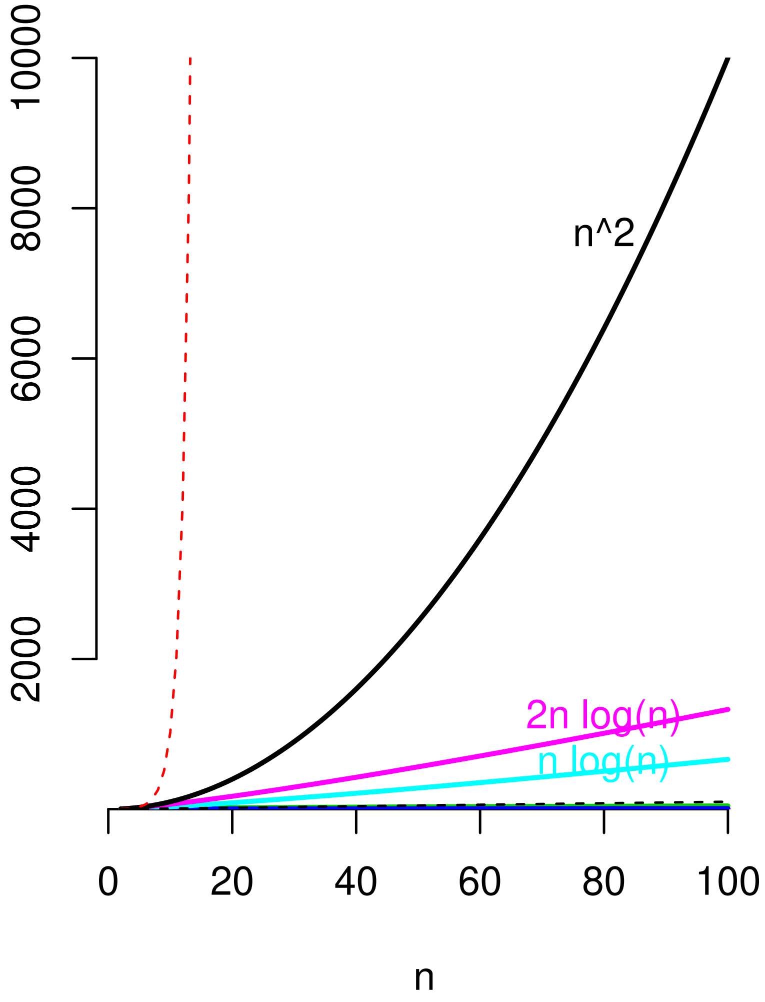
Př. 2 . Porovnání funkcí

		$n = 2$	$n = 2^{10}$	$n = 2^{20}$	$n = 2^{100}$
$\log(n!)$	$O(n \log(n))$	1	8769	19254028	$\leq 1.2710^{32}$
$(\sqrt{2})^{\log(n)}$	$\sqrt{n}$	1.4	32	1024	1.110^{15}
$2^{\log(\log(n))}$	$\log(n)$	1	10	20	100
$4^{\log(n)}$	n^2	4	1048576	1.1010^{12}	1.6110^{60}
$\sqrt{(\log(n))}$	$\sqrt{\log(n)}$	1.41	3.16	4.47	10
$n \log(n^2)$	$2n \log(n)$	4	20480	41943040	2.5410^{32}
$n \log(n)$	$n \log(n)$	2	10240	20971520	1.2710^{32}
$(\log(n))^2$	$(\log(n))^2$	1	100	400	10000

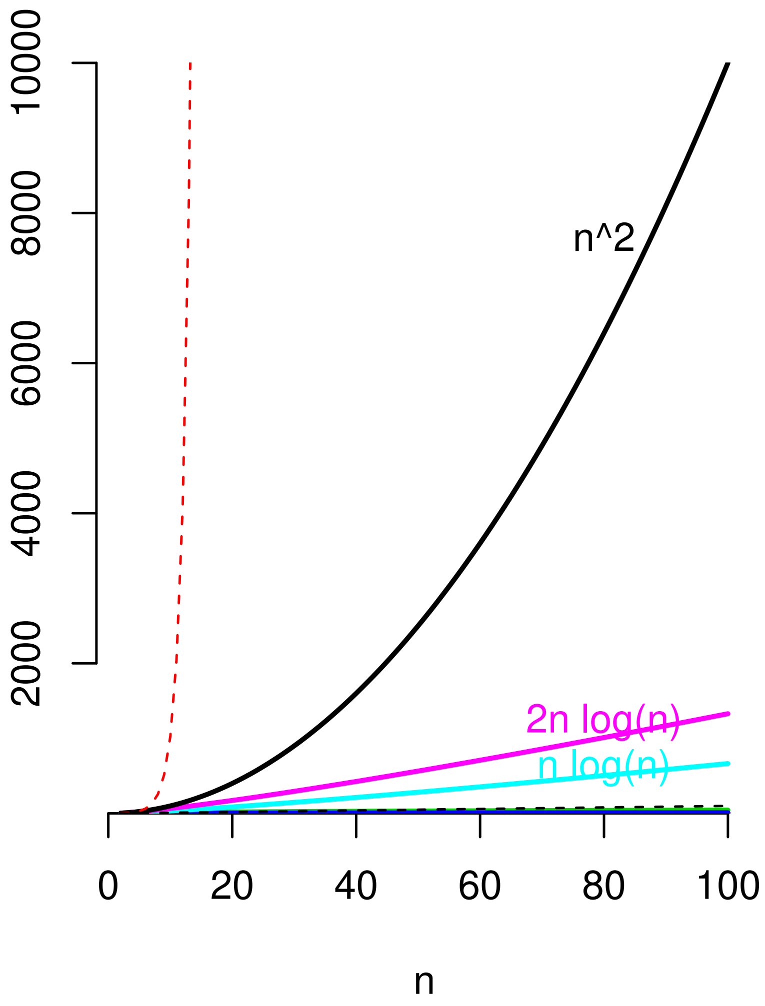
Př. 2 . Porovnání funkcí (2)

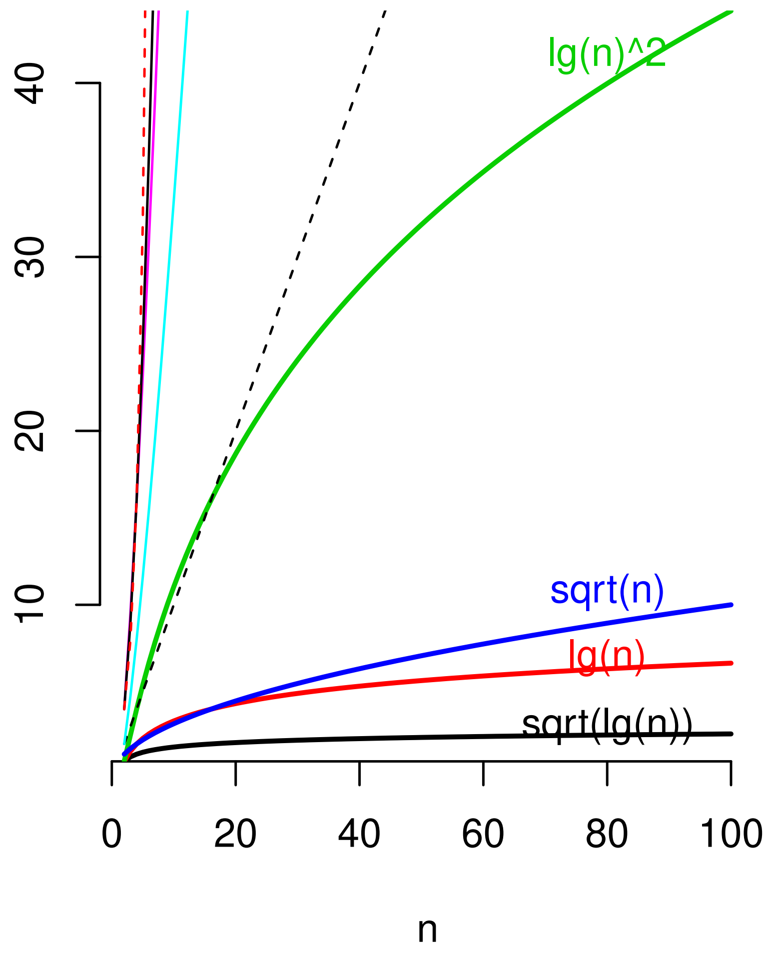


Př. 2 . Porovnání funkcí (2)

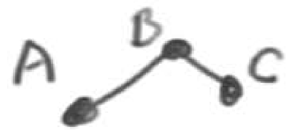


Př. 2 . Porovnání funkcí (2)





Př. 5. Převody grafových reprezentací



$G = (V, E)$, $n = |V|$, $m = |E|$

$$\begin{array}{c|ccc} & A & B & C \\ \hline A & 0 & 1 & 0 \\ B & & 0 & 1 \\ C & & & 0 \end{array}$$

I

$$\begin{array}{c|ccc} & A & B & C \\ \hline e_1 & 1 & 1 & 0 \\ e_2 & 0 & 1 & 1 \end{array}$$

II

$A \to B$
$B \to A, C$
$C \to B$

III

	I → II	II → I	I → III	III → I	II → III	III → II
$O(n)$						
$O(m)$						
$O(n^2)$						
$O(nm)$						
$O(n^2 m)$						
$O(n^2 + m)$						
$O(n^2 + nm)$						
$O(n^3)$						
$O(n^4)$						

Př. 8. Volba algoritmu

A1: $O(n\ m\ log(n))$
A2: $O((n^2 log(m))$
Který je "lepší"?

Kostry

Př. 1: násobení cen hran

(A) V grafu k ceně každé hrany přičteme stejnou nenulovou konstantou c.
(B) V grafu vynásobíme ceny všech hran stejnou nenulovou konstantou c.
V jakém vztahu budou minimální kostry původního a upraveného grafu v případech (A) a (B)? (Předpokládejte, že původní minimální kostra je určena jednoznačně.)

Př. 2: omezená kostra

Máme najít kostru grafu, nikoli nutně minimální, s tím, že je předepsáno, že cena každé hrany hledané kostry musí ležet v daném intervalu $< c1, c2 >$. Je nutno použít algoritmus pro hledání minimální kostry nebo stačí nějaký jednodušší postup?

Př. 3: zápas Prim vs. Kruskal

Uveďte asymptotickou složitost algoritmu hledání minimální kostry jednak Primova a jednak Kruskalova. Který z těchto algoritmů je asymptoticky rychlejší, za předpokladu, že počet hran grafu je čtyřnásobkem počtu uzlů?

Př. 4: hendikepovaný Kruskal

Předpokládejme, že vážený neorientovaný graf G je reprezentován svou váhovou maticí C. Určete, jaká bude asymptotická složitost Kruskalova algoritmu hledání minimální kostry za předpokladu, že doba přístupu ke každému prvku matice C je konstantní, ale zato doba každé jednotlivé operace Union i Find je vždy úměrná počtu uzlů v grafu G.

Př. 6: minimaxní kostra

V souvislém ohodnoceném grafu G s n uzly a $m = 5n$ hranami máme nalézt minimální kostru T. Poté máme z T vyrobit jinou kostru $T1$, ne už minimální, tak, že nejlacinější hranu T odstraníme z T a poté do vzniklého nesouvislého grafu přidáme nejdražší možnou hranu v G tak, aby výsledný graf byl opět stromem, tedy i kostrou. Navrhněte algoritmus řešení této úlohy a určete jeho asymptotickou složitost.

Př. 9: matice s podložkou

Předpokládejme, že graf je zadán maticí vah jednotlivých hran. Význačná hodnota v této matici (např. nekonečno, minimální/maximální hodnota číselného typu, NaN apod) indikuje, že mezi příslušnými vrcholy hrana neexistuje. Modifikujte Jarníkův-Primův algoritmus tak, aby nezávisel na počtu hran v grafu a měl složitost $O(n^2)$, kde n je počet uzlů grafu.

Př. 14: loupežník

Nenasytný loupežník odebírá z grafu co možná nejdražší hrany (uzly ponechává na místě). Nesmí ale graf rozpojit na dvě nebo více komponent, protože by byl odhalen a dopaden. Když odebere maximum všeho, co se dá, zbyde z grafu jeho minimální kostra?