1. Analýza časové složitosti nějakých pseudokódů.
2. Existuje 4-regulární graf na šesti vrcholech?
3. Existuje 1-regulární graf na šesti vrcholech?
4. Existuje graf na pěti vrcholech se souborem stupňů $2, 3, 3, 4, 4$?
5. Pro jaká $n \in \mathbb{N}$ existuje graf na n vrcholech takový, že všechny jeho vrcholy mají po dvou různé stupně?
6. Pro jaká $n \in \mathbb{N}$ existuje graf na n vrcholech takový, že všechny jeho vrcholy kromě dvou mají po dvou různé stupně?
7. Určete počet neizomorfních podgrafů na čtyřech vrcholech.
8. Jsou každé $(n-1)$-regulární grafy izomorfní?
9. Jsou každé $(n-2)$-regulární grafy izomorfní?
10. Jsou každé $(n-3)$-regulární grafy izomorfní?

1. Dokažte, že graf obsahující kružnici jako podgraf obsahuje i indukovanou kružnici jako podgraf.

2. Existuje nesouvislý graf takový, že i jeho doplněk je nesouvislý?

3. Je dán graf G na alespoň dvou vrcholech a platí, že $|E(G)| < |V(G)|$. Ukažte, že v něm musí existovat vrchol stupně nejvýše 1.

4. Určete počet všech indukovaných podgrafů grafu K_n.

5. Určete počet všech podgrafů grafu K_n.

6. Je dán graf na n vrcholech a víte, že počet komponent souvislosti je c. Ukažte, že $|E(G)| \geq n - c$.

7. Silniční síť obsahuje $2n$ měst a z každého města vede n silnic. Mezi každou dvojicí měst je nejvýše jedna silnice. Ukažte, že existuje cesta mezi libovolnou dvojicí měst.

8. Stojíte před panelákem, který má n pater. Víte, že pro nějaké patro p platí, že když hodíte vajíčko z pater $0, 1, \ldots, p$, tak se nerozbije a z vyšších patrech se rozbije. Jak najít patro p, když máte

   - 1 vajíčko,
   - nekonečně mnoho vajíček,
   - 2 vajíčka?

9. Na vstupu je pole A celých čísel délky n a číslo k. Navrhněte algoritmus, který najde dvojici čísel v poli A takovou, že jejich součet je roven číslu k. Zvládli byste to rychleji, kdyby pole bylo seřazené?

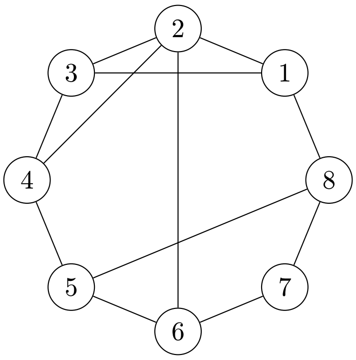
10. Modifikujte algoritmus DFS tak, aby dokázal vypsat cestu mezi vrcholy u a v, pokud existuje.

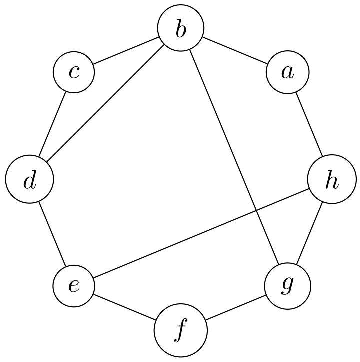
11. Na vstupu jsou dvě políčka p_1, p_2 šachovnice o rozměrech $n \times n$. Navrhněte algoritmus, který zjistí, zda-li je možné šachovým koněm proskákat z pole p_1 do pole p_2.

12. Na vstupu jsou dvě políčka p_1, p_2 šachovnice o rozměrech $n \times n$. Navrhněte algoritmus, který zjistí, zda-li je možné kulhavým koněm proskákat z pole p_1 do pole p_2. Kulhavý kůň se v lichých tazích hýbe jako kůň a v sudých tazích jako král.

1. Ukažte, že pokud jsou grafy G a H izomorfní, pak i jejich doplňky jsou izomorfní.

2. Rozhodněte, zda jsou následující grafy izomorfní.



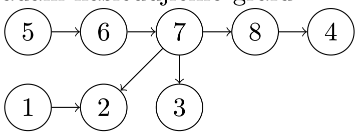


A co když se přidá hrana (a, c)?

3. Určete počet automorfismů následujících grafů.

   - Úplný graf K_n, z něhož byla odstraněna jedna hrana.
   - Úplný graf K_n, z něhož jsou odstraněny dvě sousední hrany.
   - Úplný graf K_n, z něhož jsou odstraněny dvě nesousední hrany.
   - Graf vzniklý propojením dvou úplných grafů K_n dvěma nesousedními hranami.

4. Spočítejte počet topologických uspořádání následujícího grafu



5. Definujeme $\Delta(G)$ jako maximální stupeň vrcholu grafu G. Nechť T je neorientovaný strom. Ukažte, že T má nutně alespoň $\Delta(T)$ listů.

6*. Nechť $d_1, \dots, d_n$ jsou kladná celá čísla. Dokažte, že existuje strom se souborem stupňů $d_1, \dots, d_n$, právě když $\sum_{i=1}^{n} d_i = 2n - 2$.

7. Navrhněte algoritmus, který v čase $O(n)$ pro zadaný neorientovaný graf G zjistí, zda-li obsahuje kružnici.

8*. Navrhněte algoritmus, který pro neorientovaný graf G zjistí, zda-li je bipartitní.

9. Dokažte, že pro každý souvislý graf G na alespoň třech vrcholech existují dva různé $u, v \in V(G)$ tak, že všechny grafy $G - v$, $G - u$ a $(G - u) - v$ jsou souvislé.

10. Kolik koster má $K_{2,n}$?

11. Modifikujte algoritmus *BFS* tak, aby spočítal počet nejkratších cest mezi dvěma vrcholy.

Domácí úkol: `https://codeforces.com/contest/1037` (ale prioritizujte Progtest).

1. Je dán graf G a jeho matice sousednosti $\mathbb{A}$. Jakou informaci lze vyčíst z prvku $\mathbb{A}^k_{i,j}$?

2. Je-li G souvislý graf, pak pro každou hranu $e \in E(G)$ existuje kostra, která e obsahuje. Dokažte.

3. Do cíle doběhlo n závodníků, každý si zapamatoval některé z těch, co už byli v cíli. Je možné z těchto informací jednoznačně rekonstruovat kompletní pořadí? A co kdyby nám nevadilo, kdybychom dostali dvě kompletní pořadí?

4. Hledám nejmenší z n prvků. Proč to nejde lépe než na $n - 1$ porovnání? Dokažte.

5. Jak najít v lineárním čase $\lfloor\sqrt{n}\rfloor$ nejmenších prvků z n prvků v nesetříděném poli? Dokázali byste tento problém řešit i bez binární haldy?

6. Proveďte následující operace:
   1. Build Heap na pole $14, 4, 10, 6, 2, 12, 8$.
   2. Extract min.
   3. Insert 3.
   4. Change key $14 \to 1$.

7. Mám k setříděných posloupností, které mají dohromady n prvků. Jak je rychle slít do jedné seřazené posloupnosti?

8. Jak udělat in-place HeapSort?

9*. Měli jsme setříděné pole, ale přišlo zemětřesení a každý prvek se nyní nejvýše k daleko od své původní pozice. Jak ho rychle dotřídit?

10*. Na vstupu je posloupnost délky n a kdyby se vynechalo k z nich, tak zbude setříděná posloupnost. Jak ji efektivně setřídit?

Domácí úkol: `https://codeforces.com/contest/1037/problem/D`. Deadline je do šestého cvičení.

1 Úlohy na dnešek

1. Ukažte, že lze simulovat operace *push* a *pop* fronty pomocí dvou zásobníku v amortizovaně konstantním čase.

2. Použitím penízkové metody ukažte, že operace *insert* do nafukovacího pole trvá amortizovaně konstantní čas.

3. Použitím agregované (sčítací) metody ukažte, že operaci *increment* binární sčítačky lze provést v amortizovaně konstantním čase.

4. Proveďte následující operace na minimové binomiální haldě B.

   - *Build heap* z prvků $6, 9, 11, 4$,
   - *insert* prvku 3.

   Dále vytvořte minimovou binomiální haldu B' z prvků $1, 3, 2$. Proveďte *merge* hald B a B'.

5. Jak v minimové binomiální haldě udělat operaci

   a) *decrease key*?
   b) *delete key*?
   c) *increase key*?

2 Resty

1. Je dán graf G a jeho matice sousednosti $\mathbb{A}$. Jakou informaci lze vyčíst z prvku $\mathbb{A}^k_{i,j}$?

2*. Nechť $d_1, \ldots, d_n$ jsou kladná celá čísla. Dokažte, že existuje strom se souborem stupňů $d_1, \ldots, d_n$, právě když $\sum_{i=1}^n d_i = 2n - 2$.

3. Navrhněte algoritmus na testování bipartitnosti grafu. Tzn. Váš algoritmus dostane na vstupu graf G a má rozhodnout, zda-li je graf G bipartitní.

1. Jak v binárním vyhledávacím stromu T najít k vrcholu u vrchol v s nejbližším vyšším klíčem, tzn. že hledáme vrchol v takový, že $k(u) < k(v)$ a zároveň platí, že $\forall w \in T : (k(w) > k(u) \rightarrow k(w) > k(v))$?

2. Tuhle úlohu jsme už dělali minule (v kontextu amortizované analýzy), dám ji sem jen pro úplnost. Předpokládejme, že máme ukazatel na nejmenší prvek v binárním vyhledávacím stromu T. Jak dlouho zabere $|T|$-krát hledání následníka?

3. Jak postavit vyvážený binární vyhledávací strom ze setříděného pole?

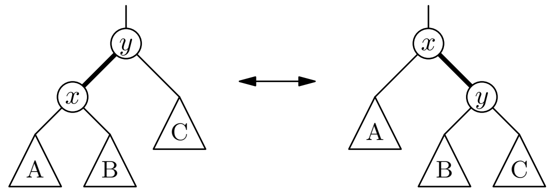
4. Navrhněte algoritmus, který v lineárním čase zadaný binární vyhledávací strom vyváží.

5. Jak projít binární vyhledávací strom s $O(1)$ pomocnou pamětí?

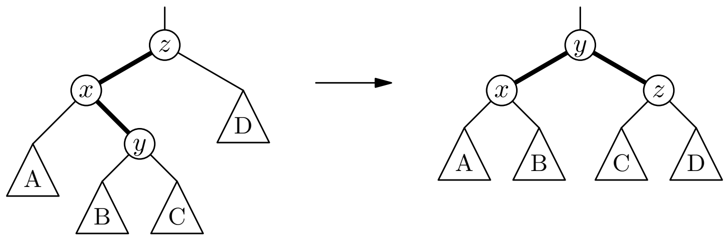
6. Proveďte na AVL stromu následující operace: *insert*(0), *insert*(32), *insert*(64), *insert*(96), *insert*(36), *insert*(−32), *insert*(56), *insert*(60), *insert*(62), *insert*(−16), *delete*(32), *delete*(56).

7*. Navrhněte algoritmus, který v lineárním čase zadaný binární vyhledávací strom vyváží s $O(1)$ pomocnou pamětí.

Domácí úkol: Udělat druhý progtest :).



Obrázek 1 Jednoduchá rotace.



Obrázek 2 Dvojitá rotace.

Zdroj obrázků: Průvodce labyrintem algoritmů od Martina Mareše a Tomáše Vally.

1. Je dáno pole velikosti $n = 7$ a chceme použít *otevřenou adresaci* s použitím *dvojitého hashování*, kde hashovací funkcí je funkce $h(k, i) = 4k + 5i \bmod n$. Vložte do tabulky prvky $8, 1, 4$, vymažte prvek 1 a nalezněte prvek 4.

2. Ukažte, že pro nesoudělná n, m platí, že $\{kn \bmod m \mid k \in \{0, \ldots, m-1\}\} = \{0, \ldots, m-1\}$.

   Hint. Použijte znalosti z BI-ZDM :).

3. Představte si, že hodíme deseti hracími kostkami a počty ok sečteme. V jakém pravděpodobnostním prostoru se tento pokus odehrává? O jakou náhodnou veličinu jde? Jak stanovit její střední hodnotu?

4. Mějme počítač, který je obohacen instrukcí `random_bit`, která dokáže náhodně uniformně nezávisle generovat nulu nebo jedničku. Jak pomocí takové funkce generovat rovnoměrně náhodná přirozená čísla od 0 do $n - 1$ pro konkrétní fixní n?

   Hint. Nejprve to zkuste pro n, které je mocninou dvojky.

5. Ze síťového socketu nám chodí packety a víme, že si je všechny nemůžeme uložit, protože se nám nevejdou do paměti. Navíc ani nevíme, kolik těch packetů přijde. Chceme si však zapamatovat alespoň jeden packet a to tak, abychom si zapamatovaný packet vybrali uniformně náhodně. Navrhněte způsob, jak tohoto docílit.

   Těžká verze. Co kdybychom si chtěli zapamatovat k uniformně náhodných packetů, tzn. každou k-tici vybrat se stejnou pravděpodobností?

   Poznámka. Tento algoritmus (resp. třída algoritmů) se jmenuje *reservoir sampling.*

6. Popište algoritmus, který v lineárním čase vygeneruje náhodnou permutaci množiny $\{0, \ldots, n-1\}$.

7. Tung letos navštěvuje na Matfyzu předmět „Pravděpodobnostní techniky“, kde každý týden dostane sadu n příkladů za domácí úkol. Tungovi to moc nejde, ale naštěstí má dobrého kamaráda Vaška, který je ochoten mu s úkoly pomáhat – každou radu však musí Vaškovi kompenzovat třemi šálky kávy. Vašek je mazaný a chce profitovat co nejvíce, ale zároveň nechce být moc nápadný. Proto na každou Tungovu žádost o radu Vašek uniformně náhodně nezávisle vybere jednu z n úloh a k té mu dá nějakou nápovědu. Jelikož je Tung opravdu marný, k jedné úloze je možné dát nekonečně mnoho nápověd. Zjistěte očekávaný počet káv, který musí Tung koupit Vaškovi, než dostane alespoň jednu radu ke každé úloze.

1. Ukažte, že algoritmus na řešení problému hanojských věží z přednášky je optimální, tzn. že pro n disků to na méně než $2^n - 1$ tahů nejde.

2. Hanojští mnichové se rozhodli, že $2^n - 1$ tahů je moc málo a zakázali si přímé přesuny disků mezi tyčemi A a B. Na kolik tahů to jde udělat teď?

3. Jak rychle dokážete třídit spojový seznam? A co když máte povoleno jen $O(1)$ pomocné paměti?

4. Dokažte, že *Karatsubův* algoritmus má lineární prostorovou složitost.

5. Představme si, že v *QuickSelectu* budeme brát jako skoromedián prvek, který leží v prostředních $\frac{3}{4}$ vstupu. Jak se změní složitost *QuickSelectu*?

6. Co kdybychom jako pivota v *QuickSelectu* volili aritmetický průměr?

7. Máme dlouhý kabel, z jehož obou konců trčí n drátů a víme, že každý drát na levém konci je propojen s právě jedním drátem na pravém konci. Nás by zajímalo, který je propojen se kterým. Jak to už bývá, dokumentace k tomuto kabelu nikdy neexistovala musíme to zjistit manuálně. K dispozici máme pouze následující operace

   (i) přivést napětí na daný drát na levém konci,

   (ii) odpojit napětí z daného drátu na levém konci,

   (iii) změřit napětí drátu na pravém konci.

   Jak pomocí těchto operací zjistíte, který drát je spojený se kterým? Jak rychle to dokážete?

8. Je dána posloupnost a_n, navrhněte algoritmus, který spočítá počet inverzí této posloupnosti.

Opakování na písemku.

1. Na vstupu je acyklický orientovaný graf (neboli DAG) s jedním zdrojem s a jedním stokem t. Navrhněte algoritmus, který spočítá počet cest mezi vrcholy s a t.

1. Navrhněte, jak efektivně třídit řetězce v čase O(součet délek). Předpokládejte, že velikost abecedy je konstantní.

2. Navrhněte, jak v čase $O(n)$ třídit celá čísla, která leží v množině $\{0, \ldots, n^3 - 1\}$.

3. Dokažte, že *extract min* v binomiální haldě nemůže být rychlejší, než $O(\log(n))$.

4. Máme matici $A \in \mathbb{N}^{n,n}$, kde každý řádek i sloupec tvoří rostoucí posloupnost. Jak rychle umíte najít prvek s hodnotou k? Čas na načtení matice nepočítejte.

Opakování na písemku.

1. Na vstupu je acyklický orientovaný graf (neboli DAG) s jedním zdrojem s a jedním stokem t. Navrhněte algoritmus, který spočítá počet cest mezi vrcholy s a t.

2. Jak rychle dokážete třídit spojový seznam? A co když máte povoleno jen $O(1)$ pomocné paměti?

3. Kolik koster mají následující grafy?
   - cesta P_n délky n,
   - kružnice C_n délky n.

4. Máte rovnoramenné váhy a 9 kuliček, z nichž je jedna těžší. Jak zjistíte, která z nich to je? A co když mám 12 kuliček?

5. Na vstupu je posloupnost délky n a kdyby se vynechalo k z nich, tak zbude setříděná posloupnost. Jak ji efektivně setřídit?

6. Tung letos navštěvuje na Matfyzu předmět „Pravděpodobnostní techniky“, kde každý týden dostane sadu n příkladů za domácí úkol. Tungovi to moc nejde, ale naštěstí má dobrého kamaráda Vaška, který je ochoten mu s úkoly pomáhat – každou radu však musí Vaškovi kompenzovat třemi šálky kávy. Vašek je mazaný a chce profitovat co nejvíce, ale zároveň nechce být moc nápadný. Proto na každou Tungovu žádost o radu Vašek uniformně náhodně nezávisle vybere jednu z n úloh a k té mu dá nějakou nápovědu. Jelikož je Tung opravdu marný, k jedné úloze je možné dát nekonečně mnoho nápověd. Zjistěte očekávaný počet káv, který musí Tung koupit Vaškovi, než dostane alespoň jednu radu ke každé úloze.

1. Je dáno n měst označených čísly $1, \dots, n$, z každého vedou cesty pouze do měst s vyšším číslem, cesty jsou jednosměrné. Kolika způsoby se lze dostat z města 1 do města n?

2. Vyřešte problém nejdelší rostoucí podposloupnosti, ve které je povolen jeden pokles.

3. Kolika možnostmi lze vydláždit mřížku velikosti $2 \times n$ dlaždičkami 1×2 a 2×1? A co čtverci 2×2, kterým chybí jeden roh?

4. Vyřešte problém nejdelší rostoucí podposloupnosti v čase $O(n \log(n))$.

5. Je dán řetězec. Jak ho rozřezat na co nejmenší počet bloků takových, že každý blok tvoří palindrom?

6. Na vstupu je vážený DAG G s jedním zdrojem sa jedním stokem t. Vymyslete algoritmus, který hledá nejkratší cestu z vrcholu s do vrcholu t.

7. Upravte algoritmus pro hledání minimální triangulace tak, aby uměl i hledanou triangulaci vypsat.

8. Na vstupu je strom G. Vymyslete algoritmus, který v něm hledá nejdelší cestu. A co když jsou hrany vážené?

9. Weighted vertex cover (WVC) (česky vážené vrcholové pokrytí)
   Vstup: graf G, cenová funkce $w : V \to \mathbb{R}_0^+$.
   Výstup: co nejlevnější podmnožina vrcholů $S \subseteq V(G)$ taková, že pro $\forall e \in E(G) \exists v \in S : v \cap e \neq \emptyset$ a $\sum_{v \in S} w(v)$ je minimální.

   Vyřešte problém WVC na váženém stromě.
   Hint: zkuste to nejdříve na neváženém stromě.
   Poznámka: na obecných grafech je WVC NP-těžké (a dokonce i nevážená je NP-težká), jde však o praktický problém a umíme jej „řešit“ nejrůznějšími způsoby. Například ho umíme 2-aproximovat, tzn. umíme najít řešení, které je nejhůře dvakrát horší, než optimální řešení.

Dynamické programování

1. Vyřešte problém nejdelší rostoucí podposloupnosti, ve které je povolen jeden pokles.

2. Na vstupu je vážený DAG G s jedním zdrojem sa jedním stokem t. Vymyslete algoritmus, který hledá nejkratší cestu z vrcholu s do vrcholu t.

3. Longest path (LPATH) (česky nejdelší cesta)
   Vstup: graf G, váhová funkce $w : E \to \mathbb{R}_0^+$.
   Výstup: cesta $P \subseteq V(G)$ taková, že $\sum_{v \in P} w(v)$ je minimální.

   Vyřešte problém LPATH na váženém stromě.
   Poznámka: problém LPATH je NP-těžký na obecných grafech, naštěstí ho řešíme jenom na stromě.

4. Weighted vertex cover (WVC) (česky vážené vrcholové pokrytí)
   Vstup: graf G, cenová funkce $w : V \to \mathbb{R}_0^+$.
   Výstup: co nejlevnější podmnožina vrcholů $S \subseteq V(G)$ taková, že pro $\forall e \in E(G) \exists v \in S : v \cap e \neq \emptyset$ a $\sum_{v \in S} w(v)$ je minimální.

   Vyřešte problém WVC na váženém stromě.
   Hint: zkuste to nejdříve na neváženém stromě.
   Poznámka: na obecných grafech je WVC NP-těžký (a dokonce i nevážená je NP-těžká), jde však o praktický problém a umíme jej „řešit“ nejrůznějšími způsoby. Například ho umíme 2-aproximovat, tzn. umíme najít řešení, které je nejhůře dvakrát horší než optimální řešení pro jakýkoliv vstup.

5. Je dán řetězec S. Vymyslete algoritmus, který v něm najde nejdelší podřetězec s, který je palindromem.

Minimální kostry

6. Ukažte, že algoritmy na hledání minimální kostry fungují i pro grafy s neunikátními vahami.

7. Máme vážený graf G a v něm jsme našli minimální kostru T. Nyní změníme váhu jedné hraně v grafu G, jak to ovlivní minimální kostru grafu G?

8. Navrhněte algoritmus, který hledá minimální kostru v grafu takovém, že váhy jsou celá čísla v rozmezí 1 až 5. Ideálně by měl pracovat v čase $O(n + m)$.

9. Jak realizovat Jarníkův algoritmus s haldou a s časovou složitostí $O(m \log(n))$?

Nejkratší cesty

10. Ukažte, jak pro libovolné n sestrojit graf na nejvýše n vrcholech takový, že mezi libovolnými dvěma vrcholy existuje $2^{\Omega(n)}$ nejkratších cest.

11. Je možné zbavit se záporných hran v algoritmech pro hledání nejkratší cesty tak, že ke všem hranám přičtu nějakou konstantu K?

12. Je dána mapa s městy a silnicemi mezi nimi. Každá cesta má nějakou délku a vybírá se na ní nějaké mýtné. Vymyslete algoritmus, který najde nejkratší cestu mezi dvěma zadanými městy takovou, že je zároveň nejlevnější.

13. Je dána mapa města, ve které jsou vyznačené mosty, které mají nějakou nosnost. Chceme převézt náklad z jednoho místa na druhé. Vymyslete algoritmus, který zjistí, jak nejtěžší náklad můžeme převézt, aby se po cestě žádný most nezhroutil.

14. V Tramtárii jezdí po železnici rychlíky, které nikde po cestě nestaví. V jízdním řádu je pro každý rychlík uvedeno počáteční a cílové nádraží, čas odjezdu a čas příjezdu. Nyní stojíme v čase t na nádraží A a chceme se co nejrychleji dostat na nádraží B. Navrhněte algoritmus, který najde takové spojení. Uvažujte, že časy jsou zadány v rámci jednoho dne s minutovou granularitou.