Každý příklad pište na zvláštní papír.

Příklad 1 (body: 3)

- Přesně definujte jednoduchý neorientovaný graf.
- Definujte kostru s minimálním hrdlem pro jednoduchý souvislý neorientovaný graf s ohodnocením hran. (nejprve definujte kostru a hrdlo)
- Co je chromatické číslo grafu?
- Uveďte jaký je rozdíl mezi řešením problému a verifikací řešení. Uveďte příklad NP-úplné grafové úlohy a demonstrujte na ní co je její řešení a co je verifikace.
- Definujte třídu NP-úplných problémů. Co je pro praxi podstatnou vlastností NP-úplných problémů?
- Definujte orientovaný acyklický graf (DAG) a uveďte pro jakou úlohu se (například) používá.
- Definujte isomorfismus grafů $G_1 = \{V_1, E_1\}$, $G_2 = \{V_2, E_2\}$. Uveďte příklad dvojice různých isomorfních a dvojice neisomorfní grafů se čtyřmi vrcholy a čtyřmi hranami.
- Co je řídký graf? Jak jej efektivně reprezentujeme v počítači?
- Definujte vzdálenostní metriku $d_G(u, v)$ prostého neorientovaného grafu.
- Jaký je v teorii her rozdíl mezi ryzí a smíšenou strategií?
- Definujte síť a tok sítí.
- Definujte sled a cestu v grafu $G = \{V, E\}$ z vrcholu A do vrcholu B.
- Přesně definujte jednoduchý neorientovaný graf.
- Definujte co je stupeň vrcholu.
- Co je to silně souvislá komponenta, pro jeký typ grafu má smysl?

Příklad 2 (body: 4)

Dokažte, že graf je strom právě tehdy pokud pro graf platí $|V| = |E| + 1$. Každou implikaci dokažte zvlášť.

Příklad 2 (body: 4)

Dokažte, že $K_{3,3}$ není rovinný graf. (vysvětlete oba pojmy, napověda: nakreslení jistého podgrafu můžete vždy převést na n-úhelník)

Příklad 2 (body: 4)

Uveďte vlastnosti relace ekvivalence a dokažte, že relace na vrcholech grafu prostého neorientovaného grafu

$$x \sim y, \text{pokud existuje cesta z vrcholu } x \text{ do vrcholu } y$$

je relace ekvivalence. Které vlastnost neplatí pro orientovaný graf? Čemu odpovídají třídy ekvivalence?

Příklad 2 (body: 3)

Definujte co je to strom. Dokažte, že graf je strom tehdy a jen tehdy, pokud v něm každé dva vrcholy spojuje právě jedna cesta.

Příklad 2 (body: 4)

Dokažte, že jakýkoliv algoritmus pro setřídění n prvkového pole jehož prvky lze pouze porovnávat, musí provést minimálně $O(n \log n)$ porovnání.

Příklad 2 (body: 4)

Definujte matici sousednosti a incidenční matici neorientovaného grafu (rozměry, hodnoty prvků). Zapište obě matice pro graf z příkladu 4. Uvažujte uspořádání vrcholů vzestupně podle písmen abecedy a hran vzestupně podle ohodnocení. Uveďte a dokažte vztah mezi maticí sousednosti, incidenční maticí a diagonální maticí stupňů vrcholů.

Příklad 3 (body: 4)

Dijkstrův algoritmus:
1) specifikujte zadání úlohy, kterou algoritmus řeší (graf, ohodnocení, počátek)
2) Popište algoritmus pomocí pseudokódu. Specifikujte potřebná vstupní data algoritmu. Výstupem algoritmu budou délky cest a odkazy na předchůdce ve stromu cest. Popište co je to prioritní fronta a jaké její operace pro algoritmus potřebujete. Implementaci prioritní fronty nemusíte diskutovat.

Příklad 3 (body: 4)

Definujte co jsou silně souvislé komponenty orientovaného grafu. Popište algoritmus hledání silně sovislých komponent včetně příslušného algoritmu na procházení grafu. Vstup algoritmu je graf popsaný řídkou maticí sousednosti, tj. pro každý vrchol máte seznam hran, které z něj vedou. Výstupem je očíslování vrcholů čísly komponent.

Příklad 3 (body: 3)

Popište pomocí pseudo kódu Kruskalův algoritmus pro hledání minimální kostry grafu s ohodnocenými hranami. Uveďte podpůrné algoritmy a datové struktury, které umožní implementaci algoritmu se složitostí $O(VlogV)$.

Příklad 3 (body: 4)

Popište pomocí pseudo kódu Primův-Jarníkův algoritmus pro hledání minimální kostry grafu s ohodnocenými hranami. Vysvětlete použití prioritní fronty pro klíčové operace.

Příklad 3 (body: 3)

Popište pomocí pseudo kódu Kruskalův algoritmus pro hledání minimální kostry grafu s ohodnocenými hranami. Uveďte podpůrné algoritmy a datové struktury, které umožní implementaci algoritmu se složitostí $O(VlogV)$.

Příklad 3 (body: 4)

Primův algoritmus:
1) specifikujte zadání úlohy, kterou algoritmus řeší (graf, ohodnocení, počátek)
2) Popište algoritmus pomocí pseudokódu (U každé operace musí být zřejmé, jak byste ji naprogramovali, zejména specifikujte použijtí operací prioritní fronty. Specifikujte potřebná

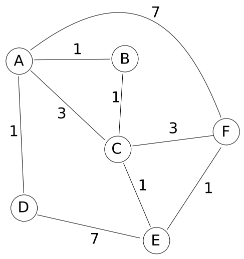
vstupní data algoritmu. Specifikujte co je výstupem algoritmu Popište co je to prioritní fronta a jaké její operace pro algoritmus potřebujete. Implementaci prioritní fronty nemusíte diskutovat.

Příklad 3 (body: 4)

Popište Ford-Fulkersonův algoritmus pro hlededání maximálního toku sítí. Vysvětlete pojmy: residuální kapacita, residuální graf, zlepšující cesta. Popište Edmons-Karp algoritmus pro hledání zlepšující cesty.

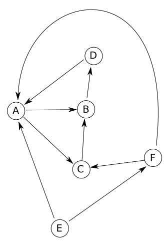
Příklad 4 (body: 3)

Demonstrujte průběh Dijkstrova algoritmu při hledání nejkratších cest z vrcholu D nakresleného grafu. Průběh algoritmu zapište do tabulky, kde jeden řádek tabulky bude obsahovat zvlášť vrcholy nenavštívené, zvlášť vrcholy ve frontě a zvlášť vrcholy s definitivní nejkratší cestou. U každého vrcholu bude aktuální délka cesty. Do grafu zakreslete strom nalezených nejkratších cest.



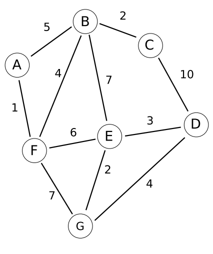
Příklad 4 (body: 3)

Demonstrujte průběh použitého algoritmu na grafu nakreslením lesů prohledávání a časy průchodů. Vstupní uspořádání vrcholů i uspořádání sousedů každého vrcholu je podle abecedy.



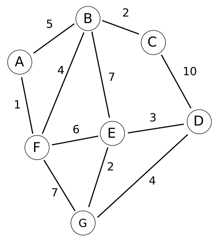
Příklad 4 (body: 3)

Demonstrujte průběh Kruskalova algoritmu na uvedeném grafu. Zapište pořadí přidávaných hran a operací $FIND$ a $UNION$.



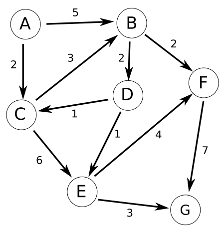
Příklad 4 (body: 3)

Demonstrujte průběh Primova-Jarníkova algoritmu na uvedeném grafu. Začněte ve vrcholu A. Pro každý krok algoritmu zapište hrany ve frontě a přidávanou hranu. Zakreslete výslednou kostru a určete její cenu.



Příklad 4 (body: 3)

Na zadané síti demonstrujte průběh algoritmu s Edmons-Karpovským hledáním zlepšující cesty. Uveďte postupné zlepšující cesty nalezené E-K algoritmem.



Příklad 5 (body: 3)

Definujte Nashovo ekvilibrium, tj. dvojici optimálních (smíšených) strategií, pro maticové hry a formulujte základní větu o maticovch hrách.

Příklad 5 (body: 4)

Popište datovou strukturu pro rozklad na disjunktní podmnožiny (komponenty) a její operace $FIND$ a $UNION$. Uveďte implementaci těchto operací pomocí stromů. Jaké dvě strategie se používají pro výrazné zlepšení amortizované složitosti?

Příklad 5 (body: 3)

Uveďte třídící algoritmus s průměrnou časovou složitostí $O(n^2)$ a algoritmus se složitostí $O(n \log n)$. Jakou časovou složitost má algoritmus průchodu grafem do hloubky v závislosti na počtu vrcholů $|V|$ a počtu hran $|E|$ grafu? Jakou průměrnou časovou složitost (při náhodném obsahu pole) má následující funkce v závislosti na velikosti pole N:

```
#define N  100
void funkce(bool pole[N])
// parametr je pole délky N předané odkazem
// jehoz prvky jsou typu boolean, tj. nabývají pouze hodnot true/false
{
   for(i=0; i<N ; i++)
     if (pole[i] == false) {
       pole[i]=true;
       break;
     } else pole[i]=false;
}
```

Co myslíte, že uvedená fukce dělá?

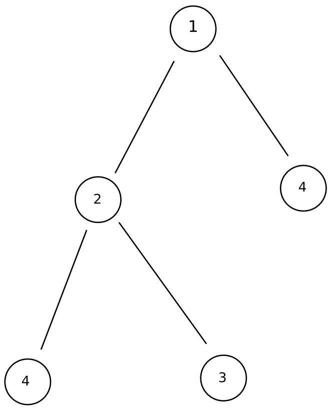
Příklad 5 (body: 3)

Popište datovou strukturu minimální binarní halda. Jaké jsou její tři vlastnosti? Jak lze n prvkovou haldu uložit v n prvkovém poli? Uveďte příklad algoritmů, které používají haldu.

Příklad 5 (body: 3)

verze A
Uveďte vlastnosti binární haldy. Na uvedeném stromu ověřte, že se jedná o (minimální)

binární haldu a proveďte operaci odeber minimum (nakreslete strom haldy po operaci).



Jak lze pomocí binární haldy implementovat prioritní frontu a jakou složitost mají operace použité v Primově algoritmu?