1. Definovat pseudopolynomiální algoritmus, h-zúžení problému, jak se liší parametrizovaný algoritmus od pseudopolynomiálních algoritmů, dokažte, že každý pseudopolynomiální alg. je max-parametrizovaný alg. (15 b.)

   Nástřel řešení a hodnocení: Uvedl jsem definice včetně parametrizace a parametrizovaného algoritmu, dále že pseudopolynomiální je speciálním případem parametrizovaných, důkaz je lehký. U parametrizace P jsem uvedl, že to je funkce z U do kladných reálných čísel (má být přirozených, i když si nejsem jist, jestli to je nevyhnutelně špatně). Mám 14 bodů.

2. Vyjádřit problém váženého párování jako úlohu celočíselného lineárního programování. (15 b.)

   Nástřel řešení a hodnocení: Problém váženého párování byl podrobně popsán u zadání – cílem je určit podmnožinu hran tak, aby žádná dvojice hran neměla společný vrchol. (Pak každá hrana určuje ten pár.) Napsal jsem tam něco ve smyslu:

   Minimalizovat: $\sum_{(u,v)\in E} x_{(u,v)} \cdot w((u,v))$

   omezení: $\sum_{(u,v)\in E} x_{(u,v)} + \sum_{(u,z)\in E} x_{(u,z)} \leq 1$ pro každé $(u,v),(u,z)\in E \times E$

   a pak že to $x_{(u,v)}$ určuje příslušnost do té hledané podmnožiny.
   Ta podmínka by měla zajišťovat to, že žádná dvojice nemá společný vrchol. Jestli si správně pamatuju pořadí příkladů, tak tam někde je možná nějakej zásek – mám 13 bodů.

3. Nadefinována úloha s náklaďáky, které odváží kontejnery různých hmotností. Každý náklaďák odveze náklad maximální hmotnosti K (všechny to stejné K). Cílem bylo vymyslet 2-aproximativní algoritmus, jak nejlépe rozložit kontejnery do aut tak, aby jich bylo potřeba co nejméně. Uvedena dokonce poznámka, ať uvážíme hladový přístup. Musel se dělat důkaz, že to tak je, a ještě popsat časovou složitost našeho algoritmu. (15 b.)

   Nástřel řešení a hodnocení: Stačilo si seřadit nakládané kontejnery od nejtěžšího a uchovávat auta, do kterých bylo něco naloženo do řady. Právě vzatý kontejner dát do auta, které je co nejdřív v řadě a vejde se do něj. Pokud takový není, vzít nový. Pak okecat, že každé auto (s možnou vyjímkou posledního) bude zaplněno z více než poloviny a tím je algoritmus triviálně 2-aproximativní.
   Složitost v $O(n^2)$ (nejhorší příklad pro složitost je, když má každý kontejner hmotnost K). Mám 15 bodů.

4. To jsem nedal: pomocí primární a duální úlohy lin programování jsme měli sestavit k-aproximativní algoritmus, který řeší problém pokrytí množin. Ten problém tam byl zase pěkně popsán. Pak jsme měli určit kolika aproximativní náš algoritmus je, a myslím, že i složitost. (15 b.)

   Nástřel řešení a hodnocení: nějak jsem to zmotal už když jsem to chtěl zapsat jako primární a duální úlohu – nejdřív jsem ztratil čas na tom, jak to vlastně standardizovat, a pak jak vyjádřit tu duální úlohu (při prvním pokusu mi vyšel nesmysl, že jsem maximalizoval něco, co nebylo shora omezený.) Definici těch úloh jsem nějak upravil, i když nevím, jestli to vůbec mohlo být dobře a to bylo všechno, co jsem tam měl. Přesto jsem dostal 5 bodů, což se trochu i divím.

5. Úkolem je najít rozklad množiny čísel na dvě podmnožiny tak, aby součet prvků v těch

podmnožinách byl co nejpodobnější. Vstupem je tedy množina čísel *M* ale je (jakkoli) rozdělena na dvě stejně mohutné podmnožiny *A* a *B* (anebo v jedné z nich je o prvek víc, když je poččet prvků lichý). Uvážíme algoritmus založený na heuristice prohledávání okolí: ten pracuje tak, že z vybere největší prvek množiny *M*, jehož přesunutím z jedné podmnožiny do druhé, se sníží rozdíl $|(\sum_{a \in A} a) - (\sum_{b \in B} b)|$. Cílem bylo analyzovat časovou složitost a najít vstupní instanci takovou, že heuristika nenajde optimální řešení, ale skončí v lokálním optimu. (10 b.)

Nástřel řešení a hodnocení: Po namalování nějaké vstupní instance a pár krocích se tak nějak odhalí, že složitost je v O(n). Stačí si uvědomit, že každý prvek bude přehozen nanejvýš jednou.
Příklad: vstup: A={1, 1}, B={2, 2} (rozdíl je 2)
algoritmus u tohoto skončí.
Optimální řešení: A={1, 2}, B={1, 2}

Po nějaké době hledání jsem nechal protipříklad A={1, 1, 1, 1, 1, 1}, B={5, 5}, protože jsem v průběhu řešení zapomněl na podmínku o mohutnosti vstupních množin. Stálo mě to jenom 2 body, což považuju taky za dost mírné (mám 8 bodů.)

6. Vstupen je *n* čísel kde každé nabývá hodnoty 0 nebo 1 a k rovnic tvaru:
   $(x_i + x_j) \ mod \ 2 = b$
   kde $i \neq j, \ i, \ j \in \{1, \ldots n\}$ a b je konstanta buď 0 nebo 1.

   Označme c^* maximální počet splnitelných rovnic. Cílem je ohodnotit literály $x_i, \ i \in \{1, \ldots, n\}$ nulami a jedničkami a to pomocí:

   1) randomizovaného algoritmu, jehož očekávaný výsledek bude splnění alespoň $\frac{1}{2}c^*$ rovnic.

   2) metodou derandomizace z něj udělat deterministický alg. Který splní alespoň $\frac{1}{2}c^*$ rovnic.
   (20 b.)

   Nástřel řešení a hodnocení: Řešením té první části je zřejmě náhodné ohodnocení. Pak pravděpodobnost, že se trefím v každé z rovnic je ½ a tudíž očekávám, že to celé je taky ½. S tím zbytkem jsem si moc neporadil. Na derandomizaci jsem se popravdě při učení už nedostal. Až ke konci času mě napadl nástin pravděpodobného řešení – když ty rovnice znám, tak můžu seřadit literály podle množství jejich výskytu, a pak začít hodnotit od toho s nejvíce výskyty. Vizi mám tu, že při ohodnocení literálu se podívám, kolik rovnic splním, bude-li literál roven jedničce, a kolik když mu přiřadéím nulu a vyberu si tu lepší možnot. Tak by to dopadlo že splním víc jak ½ rovnic a tudíž to je ok. Musela by se ale vyřešit počáteční fáze, kdy mám ve všech rovnicích obě hodnoty nepřiřazené (to by mělo být lehké). Nevím, jestli tohle řešení odpovída derandomizaci předchozího. Neměl jsem už žádný čas to tam připsat, a tak za ten chabý a neurčitý nástřel mi asi Ivanka (paní profesorka) nic nedala. Mám 8 bodů.

Hodnocení neznám, ale dost možná mohlo být klasika na procenta. Já mám 63/90, což je 70 % a mám fakt C.