**Příklad 1: Vrcholové pokrytí a aproximace**

V problému vrcholového pokrytí hledáme minimální množinu $C$ vrcholů t.ž. je incidentní se všemi hranami (alespoň jeden konec každé hranu patří do $C$.

- Ukažte, že vrcholové pokrytí patří do $NP$ (poznámka: je $NP$-úplný)
- Navrhněte polynomiální dynamický algoritmus pro stromy (poznámka: problém je polynomiální pro všechny bipartitní grafy)

Chceme aproximovat vrcholové pokrytí pro obecné grafy. Spustíme $DFS$ prohledávání a jako pokrytí vezmeme všechny vrcholy, které nejsou listy v $DFS$-stromě.

Jak dobrá je tato aproximace? Pro odhad kvality použijeme velikost vhodného párování.

**Příklad 2: Monte-Carlo integrace**

Máme dánu "spojitou hladkou" funkci $[0, 1] \to [0, 1]$. Chceme aproximovat určitý integrál této funkce.

Máme dán nepravidelný útvar (třeba mnohoúhelník) uvnitř jednotkového čtverce. Pro každý bod umíme rozhodnout zda je uvnitř tvaru nebo ne. Cheme aproximovat obsah útvaru.

**Příklad 3: Pravděpodobnostní počet různých vzorů**

Máme velkou databázi obsahující masivní množství záznamů. Každý záznam má určitý typ, který umíme určit. Chceme zhruba spočítat kolik **různých** typů se v databázi vyskytuje. Typů je ale velké množství, nedokážeme je všechny uložit do paměti.

Mějme následující algoritmus:

Zvolíme hashovací funkci $h$ mapující typy do dostatečně velkého univerza $0, \ldots, 2^i$ ($i$ je zhruba $log$(velikost databáze)). Projdeme celou databází, typ každého prvku zahashujeme a udržujeme si $k$ - nejvyšší počet koncových nul hashe, který jsme viděli. Nakonec odpovíme, že prvků je zhruba $2^k$.

Porovnějte časovou a prostorovou složitost algoritmu s přímočarým přístupem. Dokažte, že pro hashovací algoritmus existuje konstanta $p$ t.ž. s pravděpodobností $p$ je $2^k$ jedna ze dvou mocnin 2 neblíže ke skutečnému počtu vzorů v databázi.

Pro danou pravděpodobnost $\bar{p}$ navrhněte algoritmus se stejnými asymptotickými vlastnostmi, který najde nejbližší mocninu 2 s pravděpodobností alespoň $\bar{p}$.

### Domácí úkol 6: hladová aproximace MaxSAT

*MaxSAT* je optimalizační problém. Na vstupu máme formuli $\Phi$ s $n$ proměnnými, $m$ klauzulemi a celkem $|\Phi|$ literály. Ptáme kolik nejvíce klauzulí je možné současně splnit (tedy pokud $\Phi$ je splnitelná, odpověď je $m$; v opačném případě je správnou odpovědí nižší hodnota). Protože se jedná o $NP$-těžkou úlohu, chceme ji aproximovat. Pro maximalizační úlohu je algoritmus $c$-aproximace (pro $c \leq 1$) pokud nalezne řešení hodnoty apespoň $c$-násobek optima.

**Zadání:** Navrhněte hladový algoritmus běžící v čase $O(n|\Phi|)$, který zkonstruuje $\frac{1}{2}$-aproximaci řešení pro obecnou $\Phi$; a navíc pokud každá klauzule $\Phi$ obsahuje alespoň 3 různé literály, nalezne $\frac{3}{4}$-aproximaci. Zadání vyhovuje algoritmus, který v libovolném pořadí hladově fixuje hodnoty jednotlivých proměnných.

Analyzujte časovou a prostorovou složitost algoritmu. Dokažte, že algoritmus má příslušné aproximační vlastnosti a ukažte příklad klauzulí, kde algoritmus vydá právě $\frac{1}{2}$- resp. $\frac{3}{4}$-násobek optima.

Pro bonusové body můžete zkusit dosáhnout lepší časové složistosti než $O(n|\Phi|)$. Je možné dosáhnout složitosti $O(|\Phi|)$. Hint: jeden z možných přístupů je si na základě $\Phi$ vybudovat propletený systém spojových seznamů který nám umožní rychle procházet pouze klauzule obsahující danou proměnnou, a rychle mazat klauzule ze systému při částečném dosazování.

**Poznámky:** Protože maximalizueme počet splněných klauzulí, nemůžeme si formuli $\Phi$ na vstupu vhodně upravit, nemůžeme tak o $\Phi$ nic předpokládat (vyjma předpokladů v druhé části). Nemáme omezení na velikosti klauzulí, $\Phi$ může obsahovat několik stejných klauzulí, a klauzule mohou obsahovat několik stejných literálů nebo opačné literály od stejné proměnné. Se všemi těmito problémy je třeba se buď explicitně poprat, nebo navrhnout řešení tak, že se neprojeví.

**Terminologie:**

- **Literál** je výskyt proměnné, buď samotné nebo v negaci
- **Klauzule** je disjunkce několika literálů, formule je pak konjunkce klauzulí
- **Částečné dosazení** je proces resp. formule vzniklá fixováním hodnoty jedné (nebo více) proměnných. Z nové formule se smažou všechny výskyty (literály) dosazených proměnných, a všechny splněné klauzule.