Goldbergův algoritmus

Příklad 1: Pro efektivní implementaci potřebujeme u vrcholů přistupovat k nenasyceným hranám v čase $\mathcal{O}(1)$. Jak toho implementačně docílit, a jak příslušné informace udržovat při převádění přebytků?

Příklad 2: Pro efektivní implemetaci musíme umět efektivně najít vrchol s přebytkem. Navrhněte, jak implementovat a udržovat informace o takových vrcholech, abychom je mohli hledat v čase $\mathcal{O}(1)$.

Příklad 3: Pro vylepšenou implemetaci musíme najít dokonce nejvyšší vrchol s přebytkem v čase $\mathcal{O}(1)$. Vylepšete předchozí implementaci (včetně udržování při zvedání, a převodech přebytků!). Klíčové tady budou vlastnosti převádění.

Příklad 4: Připomeňte si kostru důkazu složitost Golbergova algoitmu:
Potenciál je definovaný jako součet výšek vrcholů s přebytkem.
Ukažte, že
1) Z toho, že výšky vrcholů jsou nejvýše $2n$, zvedání zvýší potenciál nejvýše o $2n^2$ za celou dobu algoritmu.
2) Z toho, že výšky vrcholů jsou nejvýše $2n$, na každé hraně proběhne nejvýše n nasycených převedení
3) nasycená převedení zvýší potenciál nějvýše o $2n^2m$ celkem
4) nenasycené převedení sníží potenciál alespoň o 1
5) nenasycených převedení je nejvýše $\mathcal{O}(n^2m)$ (potenciál je nezáporný)
6) protože všechny dílčí kroky umíme provádět v konstantím čase (viz. předchozí příklady), celková složitost algoritmu je $\mathcal{O}(n^2m)$ (inicializace apod. se ztratí)

Toky

Příklad 5: Máme děravou šachovnici (některá pole chybí). Chceme na šachovnici umístit co nejvíce věží tak, aby se navzájem neohrožovaly. Uvažujeme dvě varianty, věže se ohrožují i přes díry, a věže se neohrožují přes díry.

Příklad 6: Na vstupu máme obdélníkovou matici desetinných čísel. Navrhněte algoritmus, který všechny prvky zaokrouhlí (každý bud nahoru nebo dolu) tak, aby součty všech sloupců a řádků byly stejné jako původní součty zaokrouhlené na nejbližší celočíselnou hodnotu.

Příklad 7: Mějme množinu továren F a dolů M. Každý důl má danou cenu zprovoznění. Každá továrna potřebuje ke své produkci zprovoznit nějakou podmnožinu dolů, a potom vydává daný výdělek. Doly mohou obsluhovat libovolné množství továren. Chceme najít podmnožinu dolů t.ž. jejím zprovozněním maximalizujeme celkový výdělek.
Pro redukci úlohy lze použít následující konstrukce:
Doly a továrny budou vrcholy dvou partit, spojené hranami dle potřeb továren (neomezené kapacity). Každá partita je napojená za zdroj nebo stok, kapacity příslušných hran jsou ceny/výdělky příslušných dolů/továren.
Rozmyslete, jak lze získat řešení pomocí maximálního toku.