5. Třídicí sítě

(zapsaly T. Klimošová a K. Böhmová)

Goldbergův algoritmus – pokračování

Algoritmus upřesníme tak, že místo libovolného vrcholu s přebytkem budeme vždy pracovat s takovým vrcholem v, jehož výška $h(v)$ je největší. Jak dokážeme v následujícím lemmatu, sníží se tím počet potřebných nenasycených převedení. Takto modifikovaný algoritmus budeme značit G'.

Lemma N': V algoritmu G' je počet nenasycených převedení $\mathcal{O}(N^3)$.

Důkaz: Definujme H jako maximální výšku vrcholů s přebytkem:

$$H := \max\{h(v) : v \neq z, s, f^{\triangle}(v) > 0\}.$$

Běh algoritmu G' rozdělíme na fáze tak, že fáze skončí po každé změně H. Odhadneme počet nenasycených převedení v jedné fázi: bude jich nanejvýš stejně jako vrcholů, které se na začátku fáze nacházely na nejvyšší hladině – z jiných vrcholů v průběhu fáze nic nepřevádíme a nenasycené převedení můžeme provést z každého vrcholu nejvýše jednou. Počet nenasycených převedení za fázi je proto nejvýše N.

Odhadneme počet fází: rozdělíme fáze podle toho, jestli končí snížením nebo zvýšením H a odhadneme jejich počty. Maximálně $2N^2$ fází může končit zvýšením H, protože dle lemmatu Z provedeme nanejvýš $2N^2$ zvednutí. Snížením H může končit nanejvýš $2N^2$ fází, protože H klesne vždy alespoň o jedna, původně bylo nula a nikdy nemůže být záporné – klesnout tedy nemůže víckrát než vzrůst.

Máme maximálně $4N^2$ fází o nejvýše N nenasycených převedení, celkem tedy algoritmus G' provede $\mathcal{O}(N^3)$ nenasycených převedení. ♡

Ve skutečnosti je algoritmus ještě lepší (alespoň pro řídké grafy):

Lemma N'': V algoritmu G' je počet nenasycených převedení $\mathcal{O}(N^2\sqrt{M})$.

Důkaz: (Nezkouší se.) Algoritmus rozdělíme na fáze stejně jako v důkazu předchozího lemmatu a zvolíme přirozené číslo K (jak velké ho zvolíme, se rozhodneme až na konci důkazu). Fáze tentokrát budeme dělit na drahé, v nichž se provede více než K nenasycených převedení, a laciné, v nichž se nenasycených převedení provede maximálně K.

Nyní budeme zkoumat počty převedení v obou typech fází. Jak víme z minulého lemmatu, všech fází je nanejvýš $4N^2$, takže v laciných se provede nanejvýš $4N^2K$ nenasycených převedení. Za účelem zkoumání drahých fází definujeme potenciál:

$$\psi := \sum_{v \neq z,s;\, f^{\triangle}(x) > 0} \frac{p(v)}{K},$$

kde $p(v) := |\{u : h(u) \leq h(v)\}|$, čili počet vrcholů ve stejné nebo menší výšce než v. Víme, že na začátku bude $\psi \leq N^2/K$ a po celou dobu bude $\psi \geq 0$.

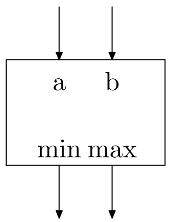
Zvednutím vrcholu v se hodnota $p(v)$ zvýší maximálně o N, u libovolného jiného vrcholu w jeho $p(w)$ klesne nebo se nezmění, potenciál tedy vzroste maximálně o N/K. Při sytém převedení po hraně z u do v (z minula víme, že se vždy provádí po hraně spádu nejvýše jedna) se mohl potenciál zmenšit o $p(u)/K$ a mohl se zvětšit o $p(v)/K$. Zvětší se tedy nanejvýš o N/K Při nenasyceném převedení z potenciálu určitě ubude $p(u)/K$ a možná přibude $p(v)/K$. Celkově se tedy sníží nanejvýš o $p(u)/K - p(v)/K$, což je $1/K$ počtu prvků na nejvyšší hladině. Z minulého lemmatu víme, že počet nenasycených převedení v jedné fázi je menší nebo roven počtu vrcholů na nejvyšší hladině na začátku fáze. Vzhledem k tomu, že zkoumáme drahé fáze, v nichž proběhne více než K nenasycených převedení, vrcholů na nejvyšší hladině musí být na začátku fáze také více než K, potenciál se tedy sníží o více než $K/K = 1$.

Potenciál ψ tedy přijme na začátku nanejvýš N^2/K, při zvedání nevzroste o víc než $(N/K)2N^2$, při sytém převádění nevzroste o víc než $(N/K)NM$, takže za celý průběh algoritmu potenciál vzroste maximálně o $(N/2)(N + 2N^2 + NM) = \mathcal{O}(N^2M/2)$, v drahých fázích tak může proběhnout nanejvýš $\mathcal{O}(N^2M/2)$ nenasycených převedení. Celkem algoritmus vykoná $\mathcal{O}(N^2K + N^2M/K)$ nenasycených převedení. Když za K dosadíme $\sqrt{M}$, dostaneme slíbený počet $\mathcal{O}(N^2\sqrt{M})$. ♡

Implementace: Narozdíl od původní verze algoritmu si ve verzi se zvedáním nejvyššího vrcholu nebudeme pamatovat seznam vrcholů s kladným přebytkem, ale setříděný seznam přihrádek. V každé přihrádce budou jen vrcholy s přebytkem s určitou výškou. Vyhledání nejvyššího vrcholu tedy zvládneme v konstantním čase, stejně pro zvýšení vrcholu nám stačí $\mathcal{O}(1)$ (buď vrchol přesuneme do vedlejší přihrádky, nebo pro něj založíme novou). Převádíme-li přebytek do vrcholu, kde předtím nebyl, pak musí mít výšku o 1 nižší, než vrchol, ze kterého přebytek převádíme (jinak by existovala nenasycená hrana se spádem dva, což nejde). Najít (případně vytvořit) přihrádku nově vzniklému vrcholu s přebytkem tak také stihneme v konstantním čase. Pro zvednutí nám tedy stále stačí čas $\mathcal{O}(N)$ a libovolné převedení přebytku zvládneme v $\mathcal{O}(1)$.

Třídění

Definice: *Komparátorová síť* je kombinační obvod, jehož hradla jsou komparátory:

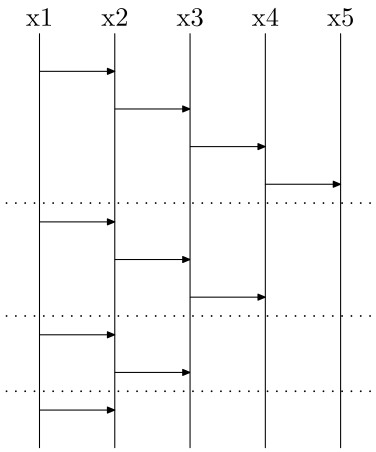


Komparátor dostane na vstupu dvě čísla, porovná je a na levý výstup vrátí menší z nich, na pravý výstup naopak vrací větší číslo ze zadané dvojice.

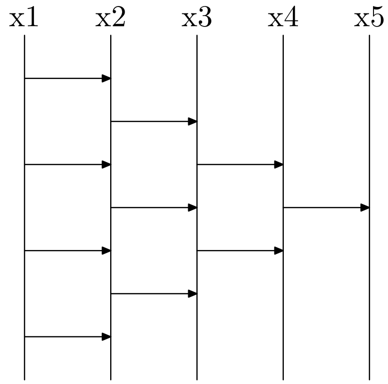
Výstupy komparátorů se nevětví.

Příklad: *Bubble sort*

Obrázek Bubble.1 ilustruje použití komparátorů pro třídění bubble sortem. Šipky představují jednotlivé komparátory.



Bubble.1



Bubble.2

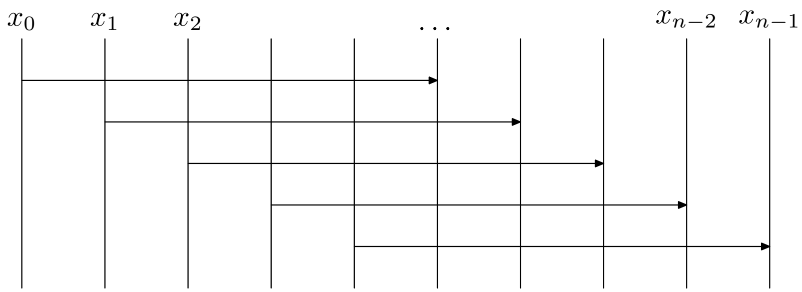
Snažíme se výpočet co nejvíce paralelizovat (viz obrázek Bubble.2). Takto se nám podařilo výpočet provést pomocí $\mathcal{O}(n^2)$ komparátorů rozmístěných na $\mathcal{O}(n)$ hladinách. Třídíme v čase n a prostoru n^2.

Definice: Řekneme, že posloupnost $x_0, \ldots, x_{n-1}$ je *čistě bitonická*, pokud pro nějaké $x_j \in \{1, \ldots, n-1\}$ platí:

$$x_0 \leq x_1 \leq \ldots \leq x_j \geq x_{j+1} \geq \ldots \geq x_{n-1}.$$

Posloupnost je *bitonická*, pokud existuje $k \in \{1, \ldots, n-1\}$, pro které je rotace původní posloupnosti o k prvků, tedy posloupnost $x_k, x_{(k+1) \bmod n}, \ldots, x_{(k+n-1) \bmod n}$, čistě bitonická.

Definice: *Separátor* S_n je síť, ve které jsou vždy i-tý a $(i+n/2)$-tý prvek vstupu (pro $i = 0, \ldots, n/2-1$) propojeny komparátorem, minimum bude i-tým, maximum $(i+n/2)$-ním prvkem výstupu.



$(y_i, y_{i+n/2}) = CMP(x_i, x_{i+n/2})$

Lemma: Pokud vstup S_n obvodu je bitonická posloupnost, pak výstup $y_0, \ldots, y_{n-1}$ je posloupnost, která splňuje:

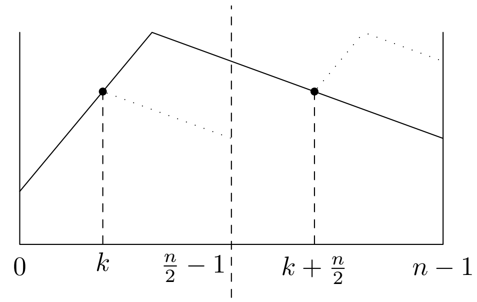
(i) $y_0, \ldots, y_{n/2-1}$ a $y_{n/2}, \ldots, y_{n-1}$ jsou bitonické posloupnosti,

(ii) Pro všechna $i, j < n/2$ platí $y_i < y_{j+n/2}$.

Důkaz: (i) Nejprve nahlédneme, že lemma platí, je-li vstupem čistě bitonická posloupnost. Tehdy najdeme nejmenší k takové, že x_k a $x_{k+n/2}$ se prohodí. (Pokud takové k neexistuje, separátor pouze zkopíruje vstup na výstup a obě tvrzení lemmatu zřejmě platí.) Řekněme, že x_m je maximum vstupní posloupnosti. Pak k bude jistě menší než m a $k + n/2$ bude větší než m, mezi k a m je tedy vstupní posloupnost neklesající, mezi $k + n/2$ a $n - 1$ nerostoucí. Uvědomíme si, že pro každé i, $k \leq i \leq n/2 - 1$ se prvky x_i a $x_{i+n/2}$ prohodí. Úsek mezi k a $n/2 - 1$ tedy nahradíme nerostoucí posloupností, první polovina výstupu tedy bude (dokonce čistě) bitonická. Úsek $k + n/2$ a $n - 1$ nahradíme čistě bitonickou posloupností, která bude neklesající, je-li $m \geq n/2$, v opačném případě je úsek $n/2 - 1$ až $k + n/2$ nerostoucí. V obou případech budou v posloupnosti maximálně dva zlomy a mezi $x_{n/2}$ a x_{n-1} bude správná nerovnost na to, aby posloupnost byla bitonická.

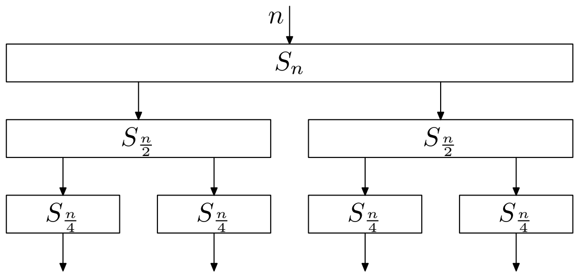
Dostaneme-li na vstupu obecnou bitonickou posloupnost, představíme si, že je to čistě bitonická posloupnost zrotovaná o r prvků (BÚNO doprava), a zjistíme, že v komparátorech se porovnávají tytéž prvky jako kdyby zrotovaná nebyla. Výstup se od výstupu čistě bitonické posloupnosti zrotovaného o r bude lišit prohozením úseků x_0 až x_{r-1} a $x_{n/2}$ až $x_{n/2+r-1}$. Obě výstupní posloupnosti tedy budou zrotované o r prvků, ale na jejich bitoničnosti se nic nezmění.

(ii) Z důkazu (i) pro čistě bitonickou posloupnost víme, že $y_0 \ldots y_{n/2-1}$ čistě bitonická a bude rovna $x_0 \ldots x_{k-1}, xk + n/2 \ldots x_{n-1}$ pro vhodné k a navíc bude mít maximum v x_{k-1} nebo $x_k + n/2$. Mezi těmito body ovšem ve vstupní posloupnosti určitě neležel žádný x_i menší než $x_k - 1$ nebo $x_k + n/2$ (jak je vidět z obrázku) a posloupnost $x_k \ldots x_{k-1+n/2}$ je rovna $y_{n/2} \ldots y_{n-1}$. Pro obecné bitonické posloupnosti ukážeme stejně jako v (i). ♡



······ posloupnost prohozená separátorem

Definice: *Bitonická třídička* B_n je obvod sestavený ze separátorů, který dostane-li na vstupu bitonickou posloupnost délky n (konstruujeme třídičku pro $n = 2^k$), vydá setříděnou posloupnost délky n.

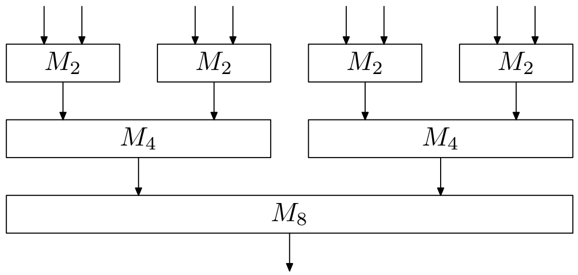


Bitonická třídička B_n

Separátor má jednu hladinu s $\mathcal{O}(n)$ hradly, třídička tedy bude mít $\log n$ hladin s $\mathcal{O}(n \log n)$ hradly.

Příklad: *Merge sort*

Bitonická třídička se dá použít ke slévání setříděných posloupností. S její pomocí sestavíme součástky mergesortové sítě. Setříděné posloupnosti $x_0, \ldots, x_{n-1}$ a $y_0, \ldots, y_{n-1}$ spojíme do jedné bitonické $x_0, \ldots, x_{n-1}, y_{n-1}, \ldots, y_0$. Z takové posloupnosti pomocí B_{2n} vytvoříme setříděnou posloupnost. Blok M_{2n} sestává z bloku B_{2n}, jehož druhá polovina vstupů je zapojena v obráceném pořadí.



Z bitonických třídiček tedy můžeme postavit mergesortovou síť, která bude mít $\mathcal{O}(\log^2 n)$ hladin a $\mathcal{O}(n \log^2 n)$ hradel.

Existuje třídicí algoritmus, kterému stačí $\mathcal{O}(\log n)$ hladin, ale jeho multiplikativní konstanta je příliš veliká, takže je v praxi nepoužitelný.