

Navazování spojení

Řešení problému

Potřebujeme přidat co nejméně hran tak, aby bylo možné projít všechny vrcholy grafu.

Zpracování vstupu

Abychom neprocházeli zbytečně moc možností vezmeme si ze vstupu jen to, co potřebujeme. Chceme získat páry vrcholů, Hlavu (H) a Ocas (T), pro které platí, že z hlavy se dostaneme do Ocasu, do Hlavy nesměruje žádný jiný vrchol a z Ocasu nevede žádná hrana v případě, že se nejedná o samostatný vrchol či cyklus.

V mém popisu využívám číslované vrcholy. Nemusí to být čísla, může to být absolutně cokoliv, čísla využívám pro jednodušší vysvětlení.

Tyto páry jsou ku příkladu v ukázkovém grafu: 1H-4T, 6H-4T, 6H-7T. V případě, že jsme narazili na cyklus tak se nic nemění a bude to fungovat stejně.

U rozšíření: 1H-4T, 8H-T, 5H-5T, v tomto příkladě máme cyklus (5H-5T).

Co budeme využívat?

Vytvoříme si list na Hlavy (HL), Ocas (TL), list Unikátních Párů hlav a ocasů (SL) a slovník Vrcholů (VL). HL a TL budou držet čísla vrcholů, SL drží vždy 2 čísla, indexy k listu hlav a ocasů.

Jak zjistíme Hlavy a Ocas?

Začneme na jakémkoliv vrcholu, vždy se podíváme na vrcholy s hranami směřujícími z našeho současného vrcholu a vrcholů, ze kterých můžeme jít do vrcholu, na kterém zrovna jsme. Vždy, když začínáme na prvním vrcholu bez toho, abychom se do něj dostali pokračováním, vytvoříme nové 2 čísla „indexy“, HI „Head index“ a TI „Tail index“, které budou současné délky HL a TL, například když začínáme s prázdnými listy na jakémkoliv vrcholu, délky obou listu HL i TL jsou 0, takže naše současné HI a TI budou 0 a 0, v případě že už máme HL [6,7,5,1] a TL [0,5], tak naše nově vytvořené HL bude 4 a TL 2. Vždy na každém nově objeveném vrcholu provedeme „**Objevení Vrcholu**“ s drženými indexy. Po Objevení vrcholu se vždy prvně podíváme, kam všude můžeme jít, přednost ale vždy má především cesta zpátky (hrana směřující do našeho současného vrcholu), ostatní možné cesty provedeme po této, ale vždy bude mít cesta zpátky přednost. V případě, že potkáme více cest zpátky, tak jedna pokračuje beze změn a další si vytvoří nový HI, pokud je více cest vpřed, jedna pokračuje beze změn a další si vytvoří nový TI. Pokaždé provedeme „Uložení vrcholu“. Nezačínáme na již prozkoumaných vrcholech, pokud jsme prozkoumali všechny vrcholy uděláme z SL list reálných párů, a ne jenom referencí. Reálné hodnoty získáme tak, že projdeme všechny SL, například $SL_i = SL[0]$ a vždy vezmeme $SL_i[0]$ a použijeme jako index v HL a $SL_i[1]$ jako index v TL, díky kterým získáme reálné hodnoty z HL a TL, tím získáme jeden pár. Získáme všechny reálné hodnoty a ty vrátíme, tím končíme se „**Zpracováním vstupu**“.

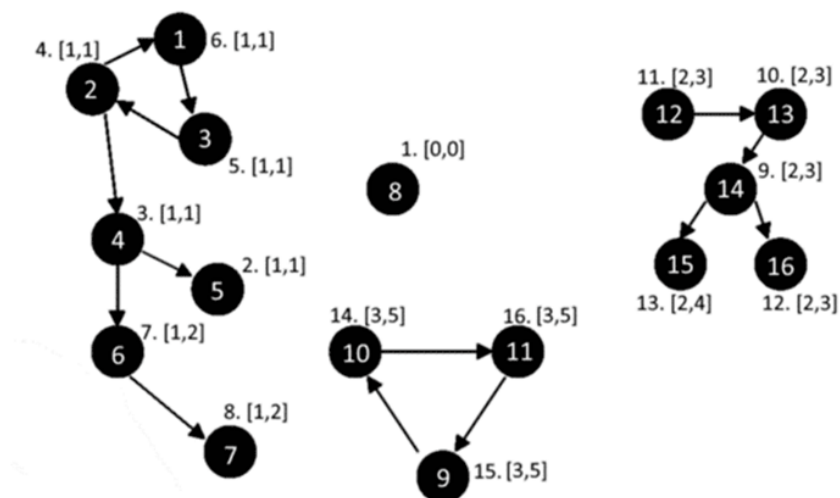
Objevení vrcholu

V případě, že nemáme ještě vytvořená čísla v HL a TL vytvoří jak Hlavu, tak Ocas jako současný vrchol, v případě že už jsou vytvořená čísla na indexech, přepíše se číslo v TL na pozici TL na číslo současně objeveného vrcholu. Dále, jestli objevíme vrchol, který již byl objevený, a pokud je objevený bod ocas našeho současného vrcholu tak si ho nastavíme jako i jako hlavu (rozšíření – cyklus). Pokud je objevený ale není to i náš ocas, tak si vezmeme a) jeho hlavu (jdeme opačným směrem hran), b) ocas (jdeme po směru hran) a pokud se to stane, dále nehledáme, pokud není jiná hrana ze současného vrcholu, která směřuje jinam.

Uložení vrcholu

Přidá do SL, „listu Unikátních Párů“, pokud tam ještě není.

Tady je ukázka, jak najít hlavy a ocasy, tato ukázka pokrývá i rozšíření. U každého vrcholu je uvedeno číslo, které reprezentuje pořadí kroku, ve kterém jsme ho prozkoumali. a dole Seznam kroků pod schématem ukazuje, jak vypadaly listy po provedení daného kroku.



1. HL [8] TL [8] SL [[0,0]]
2. HL [8,5] TL [8,5] SL [[0,0],[1,1]]
3. HL [8,4] TL [8,5] SL [[0,0],[1,1]]
4. HL [8,2] TL [8,5] SL [[0,0],[1,1]]
5. HL [8,3] TL [8,5] SL [[0,0],[1,1]]
6. HL [8,1] TL [8,5] SL [[0,0],[1,1]]
7. HL [8,1] TL [8,5,6] SL [[0,0],[1,1],[1,2]]
8. HL [8,1] TL [8,5,7] SL [[0,0],[1,1],[1,2]]
9. HL [8,1,14] TL [8,5,7,14] SL [[0,0],[1,1],[1,2],[2,3]]
10. HL [8,1,13] TL [8,5,7,14] SL [[0,0],[1,1],[1,2],[2,3]]
11. HL [8,1,12] TL [8,5,7,14] SL [[0,0],[1,1],[1,2],[2,3]]
12. HL [8,1,12] TL [8,5,7,16] SL [[0,0],[1,1],[1,2],[2,3]]
13. HL [8,1,12] TL [8,5,7,16,15] SL [[0,0],[1,1],[1,2],[2,3],[2,4]]
14. HL [8,1,12,10] TL [8,5,7,16,15,10] SL [[0,0],[1,1],[1,2],[2,3],[2,4],[3,5]]
15. HL [8,1,12,9] TL [8,5,7,16,15,10] SL [[0,0],[1,1],[1,2],[2,3],[2,4],[3,5]]
16. HL [8,1,12,10] TL [8,5,7,16,15,10] SL [[0,0],[1,1],[1,2],[2,3],[2,4],[3,5]]

Po posledním kroku list SL obsahuje indexy vrcholů v listech HL (Hlavy) a TL (Ocasy), které tvoří hledané páry, např. poslední prvek v SL ([3,5]) odkazuje na vrchol s indexem 3 v HL (vrchol 9) a vrchol s indexem 5 v TL (vrchol 10). Z této ukázky dostaneme 6 párů: 8H-8T, 1H-5T, 1H-7T, 12H-16T, 12H-15T, 10H-10T.

Úprava zpracovaného vstupu

Ve chvíli, kdy máme tyto páry, odstraníme duplikáty. Tím je myšleno, že pokud Hlava i Ocas již existují jako Hlava/Ocas u jiného páru, tento pár vymažeme.

Například u ukázkového grafu odstraníme 6H-4T, protože 6H a 4T se objevily v jiném páru. Tento pár tedy odstraníme a dostaneme 1H-4T, 6H-7T. Jak to ale uděláme? Abychom je vymazali, projdeme nejdříve všechny páry a uděláme si dva slovníky, kde uložíme počty vrcholů v Hlavách a v Ocasech. Pokud délka slovníků odpovídá počtu vrcholů, nic neodstraňujeme, ale pokud ano, zkontrolujeme, zda jsou duplikáty a odstraníme duplikáty, pokud ano.

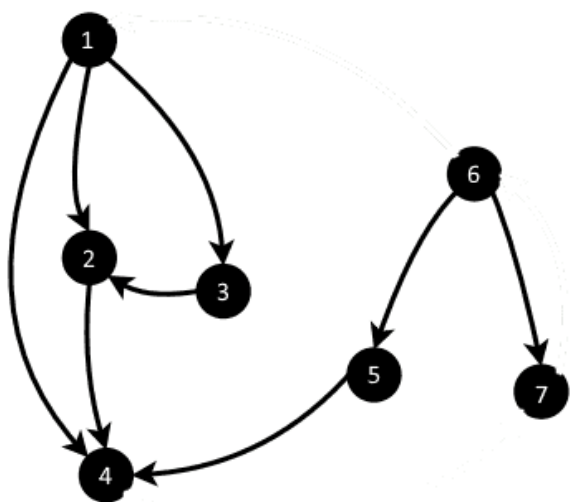
U rozšíření žádné duplikáty nemáme, takže nám zůstalo: 1H-4T, 8H-8T, 5H-5T.

Vytvoření hran

Ted' už jen poslední krok – propojíme Ocas směrem do hlavy, která se nachází o jeden pár dále. Pokud je jen jeden pár, pak to bude samo se sebou a pokud je to poslední pár, propojíme Ocas s první Hlavou. U ukázkového grafu jsme měli 1H-4T a 6H-7T a z toho se nám stanou hrany 1H <- 7T a 6H <- 4T. Pokud nastane například 1H <- 1T nebo pokud se vytvoří již existující hrana, tak nepřidá se, protože nechceme dát hranu samu k sobě či duplikovat hrany.

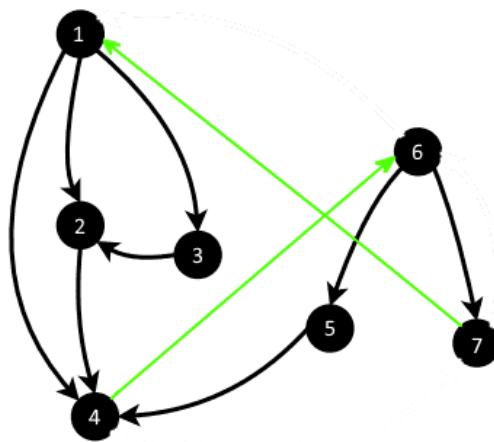
U rozšíření získáme hrany 1H <- 5T, 8H <- 4T, 5H <- 8T.

Ukázkový graf

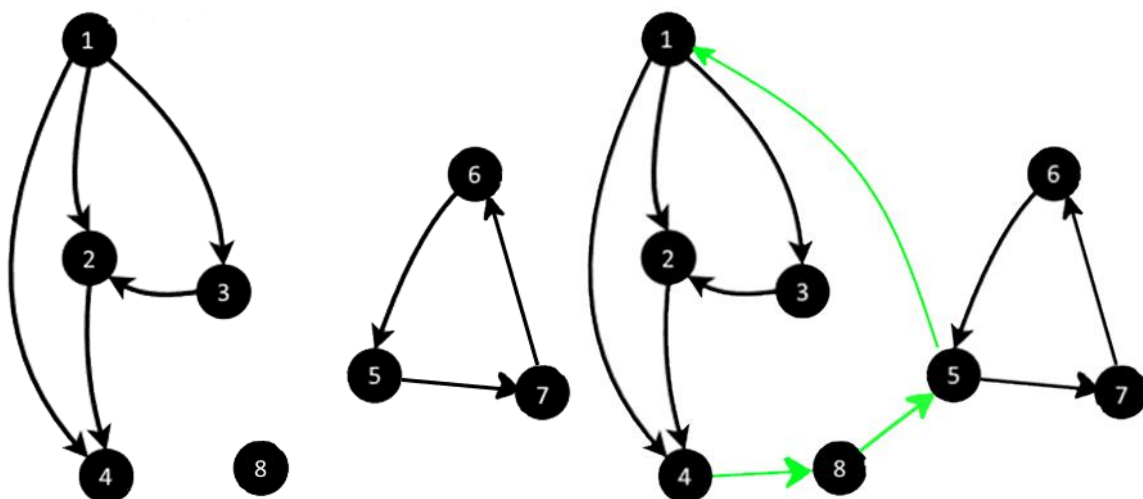


(Rozšíření) Ukázkový graf

Řešení ukázkového grafu



(Rozšíření) Řešení ukázkového grafu



Zdůvodnění řešení

Toto řešení bude vždy funkční, protože jediné vrcholy, které potřebujeme propojit jsou ty, které nemají jak vstupní, tak odchozí hrany (nepočítaje cyklus). Proto nám stačí znát Hlavy a Ocasy a pokud propojíme každý Ocas s následující Hlavou, vznikne cyklus.

Díky tomu, že odstraňujeme duplikáty, dostaneme i řešení s nejméně možnými hranami, protože nechceme propojit tyto hrany více než jednou, tudíž řešení s méně hranami nejde dosáhnout.

Zdůvodnění nejlepšího řešení.

Lepší řešení než moje s časovou komplexitou $O(n)$ být nemůže, protože to by znamenalo že ani nezpracují vstup, a tudíž by neměli dost informací k udělení nejlepšího řešení. Nemůžeme říct jaký bod spojit s jakým, bez procházení všech vrcholů a zjišťováním jaké body budeme potřebovat spojit a jaké ne. To by bylo to samé jako zjistit jaké dveře otevřít, bez vědomí, jaké jsou dveře a kam vedou bez žádných dalších informací, bylo by to o náhodě.

Časová komplexita

Získání Hlav a Ocasů ze vstupu je časová komplexita $O(n)$. Odfiltrování duplikátů je $O(k)$ kde k ($0 \leq k \leq 2n$) představuje počet Hlav a Ocasů. Časová komplexita této funkce je $O(n)$. Poslední část, vytvoření hran, je $O(p)$ kde p ($0 \leq p \leq n/2$). Časová komplexita této funkce je $O(n)$.

Tyto funkce jsou vykonány sekvenčně, celková časová komplexita tohoto algoritmu je tedy $O(n)$.