Grafové algoritmy

2009/10, 1. termín

1. Maximální párování

Je dán neorientovaný graf $G = (V, E)$. Množina $M \subseteq E$ je *párování*, pokud žádné dvě různé hrany z M nemají společný vrchol. *Maximalita* párování se rozumí vzhledem k počtu hran.

Vrchol $v \in V$ je *volný* vzhledem k párování M, není-li koncem žádné z hran v M. Cesta $P = (v_0, v_1, \ldots, v_k)$, $k \geq 1$, v G je *střídavá* vzhledem k párování M, jsou-li vrcholy na ní po dvou různé a střídají-li se v ní hrany z M a $E \setminus M$. Taková cesta je *volná* vzhledem k M, jsou-li oba její konce volné. Její *alternací* dostáváme párování M', které se od M liší právě tím, že pro libovolnou hranu e na P je $e \in M$ právě když $e \notin M'$ a naopak.

a) Dokažte větu:

Věta. M je maximální párování v G právě když v G neexistuje volná cesta vzhledem k M.

Pozn. V důkazu netriviální implikace můžete uvážit dvě párovaní M a M' a diskutovat jak vypadají souvislé komponenty grafu $(V, M \cup M')$.

Ve zbytku pojednání je graf G *bipartitní*, tj. existují neprázdné disjunktní množiny vrcholů X, Y dávající ve sjednocení celé V a takové, že libovolná hrana z E ma jeden konec v X a druhý v Y. Fixujme taková X, Y.

V následujícím algoritmu je
$Match[y] = x$, je-li $\{x, y\} \in M$ a
$Match[y] = 0$, je-li y volný vzhledem k M.

Jeho podstatou je hledání volných cest z $u \in X$ pomocí BFS(G). Volné cesty z $v \in Y$ nemusíme uvažovat (stačí je opačně orientovat). Pro cestu $(u, v, u', v', u``, v'', \ldots)$ klademe $Prev[v] = u,\ Prev[v'] = u', \ldots$. Navštívené vrcholy z X máme v poli $Q[1], Q[2], \ldots$ a ty z Y množině N.

b) V uvedených algoritmech doplňte řádky 18 resp. 4.

```
MaxMatching(G)
1  for i ← 1 to n
2      Match[i] ← 0
3  for u ∈ X
4      Q[1] ← u
5      Qsize ← 1
6      for i ← 1 to n
7          Prev[i] ← 0
8      k ← 1
9      repeat
10         x ← Q[k]
11         for y ∈ Adj[x]
12             if y ∉ N
13                 přidej y do N
14                 Prev[y] ← x
15                 if y je volny
16                     Alternuj(y)
17                     goto 1
18                 přidej ......... do Q
19         k ← k + 1
20     until k > Qsize
21     odstraň z grafu Q a N
22     1:
```

```
Alternuj(y)
repeat
    w ← Prev[y]
    Match[y] ← w
    ...
    Match[w] ← y
    y ← v
until y = 0
```

c) Doplňte důkaz věty :

Věta. Množina $M = \{\ \{x, Match[x]\} \mid x \in X,\ Match[x] \neq 0\ \}$ je maximálním párováním grafu G.

Důkaz. Pokud žádný z X nezůstane volný, máme maximální párování. Pokud z $u \in X$ najdeme volnou cestu, alternujeme ji. Pokud z $u \in X$ existuje volná cesta, najdeme ji, neboť před ř. 21 máme v Q vrcholy právě vrcholy z X dosažitelné z u pomocí

.... začínající hranou ....

a v množině N ....

Přitom platí

(i) $|Q| = |N| +$ ....

(ii) každý vrchol z N je zpárován s ....

(iii) neexistuje hrana mezi Q a $Y \setminus N$.

Zbývá ukázat, že vrcholy z Q a N a s nimi incidentní hrany je možné „beztrestně“ odstranit.

I kdyby jsme je neodstranili, nemůže v budoucnu vést volná cesta z $u' \in Q$, $u' \neq u$, neboť

....

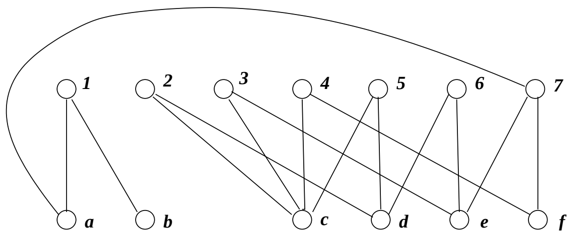
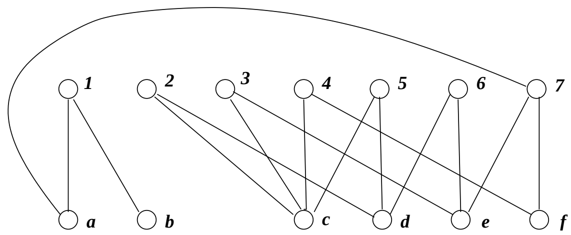
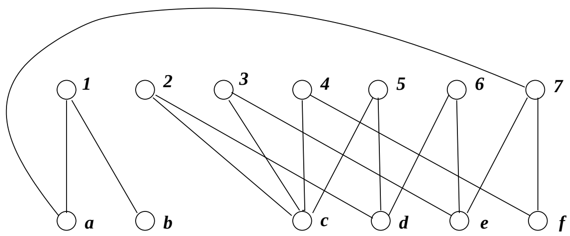
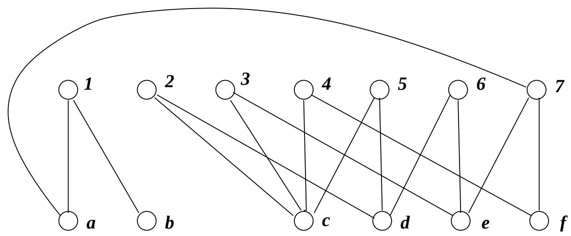
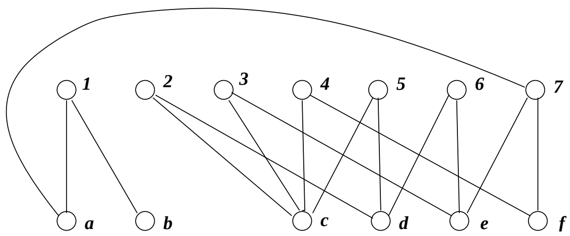
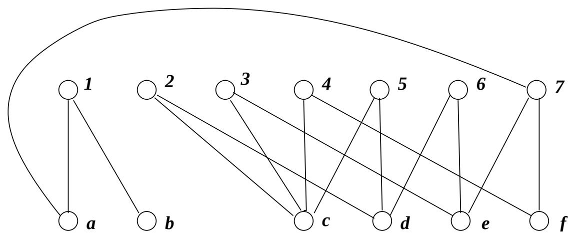
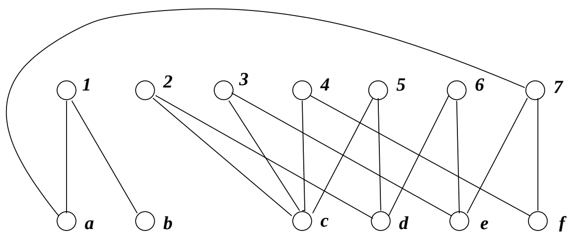
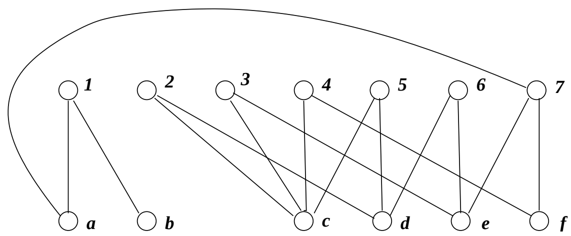
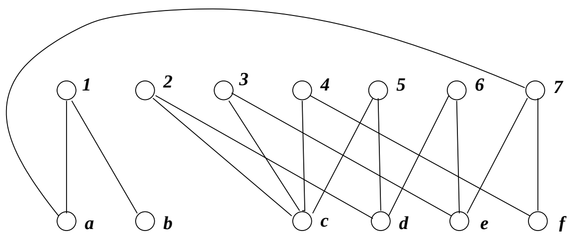
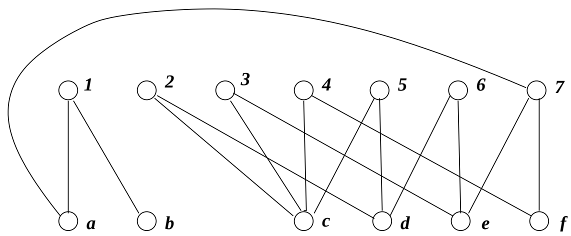
Rovněž, kdyby taková cesta vedla z $u' \in X \setminus Q$ a nezůstala celá v $X \setminus Q$ a $Y \setminus N$, navštívila by nejprve nějaké $v \in N$ před tím než by mohla do Q kvůli .... Pak by pokračovala do

....
a poté do

...., atd.,
a nemohla by skončit v nějakém volném vrcholu kvůli

....
- SPOR

d) Do přiložených diagramů vyznačte průběch algoritmu. Hrany aktuálního párování značte červeně, orientované hrany $(y, Prev[y])$ značte modře a odstraněné vrcholy a hrany budou černě (pozor : odstraněná hrana může být v párování). Nový obrázek kreslete, kdykoliv proběhne některý z řádků 7,16,21. Můžete vynechat malování pro $u = 1, \ldots, 5$ a začít s $u = 6$. Seznamy sousedů jsou seřazeny podle velikosti či abecedy. Rovněž výběr u probíhá podle velikosti. Vyznačujte též $Q[1], Q[2], \ldots, Q[Qsize]$ a vrcholy $y_1, y_2, \ldots$ z řádku 13.


1 2 3 4 5 6 7
a b c d e f

1 2 3 4 5 6 7
a b c d e f

1 2 3 4 5 6 7
a b c d e f

1 2 3 4 5 6 7
a b c d e f

1 2 3 4 5 6 7
a b c d e f

1 2 3 4 5 6 7
a b c d e f

1 2 3 4 5 6 7
a b c d e f

1 2 3 4 5 6 7
a b c d e f

1 2 3 4 5 6 7
a b c d e f

1 2 3 4 5 6 7
a b c d e f