Примерни решения на домашно № 2, 28 май 2023 г.

Задача 1: Нека $\{a_1, a_2, \ldots\}$ е изброимо безкрайно множество от върхове. Нека c също е връх. За всяко $n \in \mathbb{N}^+$, нека D_n е графът

$$D_n = (\{a_1, \ldots, a_n, c\}, \{(a_1, a_2), (a_2, a_3), \ldots, (a_{n-1}, a_n), (a_1, c), (a_2, c), \ldots, (a_n, c)\})$$

1 т. 1. Нарисувайте D_1, D_2 и D_3.

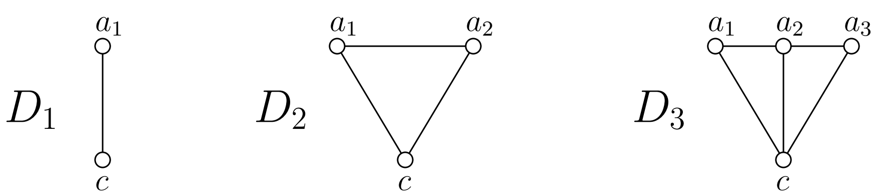
1 т. 2. Предложете индуктивна дефиниция за множеството $\mathcal{D} = \bigcup_{n \in \mathbb{N}^+} \{D_n\}$. Няма нужда да доказвате еквивалентността на двете дефиниции.

2 т. 3. Нарисувайте всички покриващи дървета на D_1, на D_2 и на D_3. За да е напълно ясна рисунката, първо намерете колко покриващи дървета (като именувани графи!) има всеки от D_1, D_2 и D_3, после нарисувайте толкова негови копия и след върху всяко копие с цветно нарисувайте съответното покриващо дърво.

26 т. 4. Нека S_n е броят на покриващите дървета на D_n. Намерете хомогенно линейно рекурентно уравнение с константни коефициенти от втори ред за S_n.

10 т. 5. Решете това уравнение.

Решение: Ето D_1, D_2 и D_3:



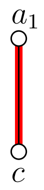
Индуктивна дефиниция може да е следната. Базовото множество се състои от едно единствено ребро $f = (a, b)$. Графът $(\{a, b\}, \{f\})$ е в $\mathcal{D}$. Казваме, че f е *важното ребро* в този граф и връх a е *важният връх* в този граф.

Нека е даден произволен граф $H \in \mathcal{D}$, като важното ребро на H е реброто $e = (u, v)$ и важният връх на H е u. Нека w е връх, който не е връх в H. Тогава графът

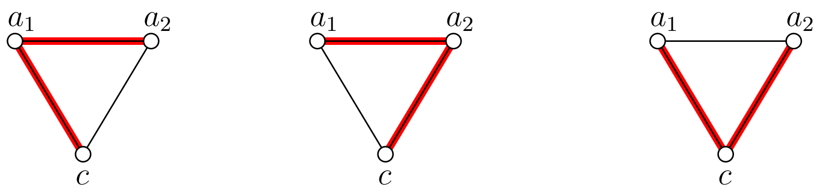
$$H' = (V(H) \cup \{w\}, E(H) \cup \{(u, w), (v, w)\})$$

е от $\mathcal{D}$, като важното ребро на H' е (u, w) и важният връх на H' е u.

Да нарисуваме покриващите дървета. D_1 има точно едно покриващо дърво:

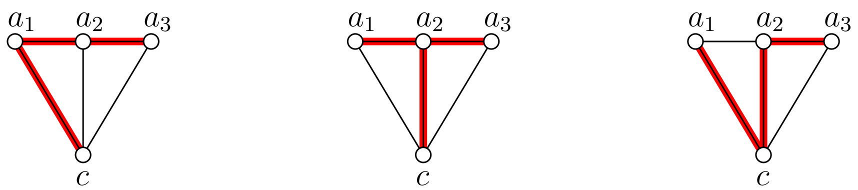


D_2 има точно три покриващи дървета (точно едно ребро на D_2 не участва):

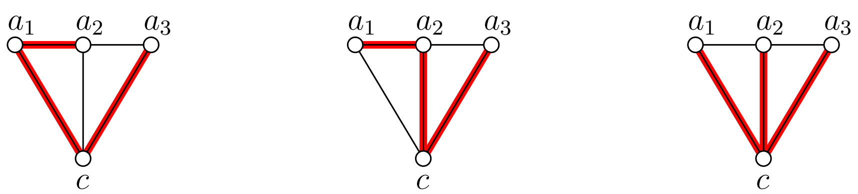


D_3 има точно осем покриващи дървета, но е добре да ги генерираме систематично, за да сме сигурни, че не сме изпуснали. Съгласно индуктивната дефиниция, можем да мислим за D_3 като състоящо се от D_2 с важно ребро (a_2, c) и важен връх c, с добавяне на нов връх a_3, нови ребро (a_2, a_3) и (a_3, c), като важният връх продължава да е c, а важното ребро става (a_3, c). Разбиваме покриващите дървета на D_3 на следните три множества.

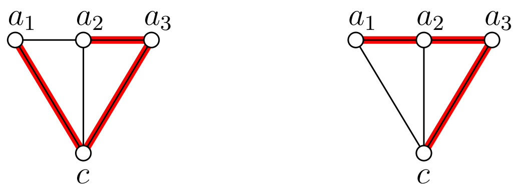
- Първо, покриващите дървета на D_3, които се състоят от едно покриващо дърво на D_2 и реброто (a_2, a_3):



- Второ, покриващите дървета на D_3, които се състоят от едно покриващо дърво на D_2 и реброто (a_3, c):



- Трето, покриващите дървета на D_3, които съдържат както реброто (a_2, a_3), така и реброто (a_3, c). Но щом тези две ребра присъстват, останалите ребра не може да образуват покриващо дърво на D_2, защото би имало цикъл. Като първи подслучай разглеждаме покриващото дърво, което се състои от (a_2, a_3), (a_3, c) и покриващото дърво на D_1. Като втори подслучай разглеждаме покриващото дърво, което се състои от (a_2, a_3), (a_3, c) и (a_1, a_2):



Можем да обобщим тези резултати така. Множеството от покриващите дървета на D_n се разбива на следните множества.

- Покриващите дървета на D_{n-1} плюс реброто (a_{n-1}, a_n). Мощността на това множество е S_{n-1}.
- Покриващите дървета на D_{n-1} плюс реброто (a_n, c). Мощността на това множество е S_{n-1}.
- Покриващите дървета на D_{n-2} плюс ребрата (a_{n-1}, a_n) и (a_n, c). Мощността на това множество е S_{n-2}.
- Покриващите дървета на D_{n-3} плюс ребрата (a_{n-2}, a_{n-1}), (a_{n-1}, a_n) и (a_n, c). Мощността на това множество е S_{n-3}.
- И така нататък.
- Покриващите дървета на D_1 плюс ребрата (a_2, a_3), (a_3, a_4), $\ldots$, (a_{n-1}, a_n) и (a_n, c). Мощността на това множество е S_1.
- Покриващото дърво, състоящо се от ребрата (a_2, a_3), (a_2, a_3), (a_3, a_4), $\ldots$, (a_{n-1}, a_n) и (a_n, c). То е само едно.

Оттук заключаваме, че

$$S_n = \begin{cases} 1, & \text{ако } n = 1 \\ 2S_{n-1} + S_{n-2} + S_{n-3} + \cdots + S_1 + 1, & \text{ако } n \geq 2 \end{cases} \tag{1}$$

Рекурентното уравнение (1) обаче не е от втори ред и дори не е с крайна история, така че не може да го решим с метода, изучаван на лекции. Но можем да го преобразуваме в еквивалентно уравнение от втори ред така. За всяко достатъчно голямо n е в сила

$$\begin{aligned} S_n &= 2S_{n-1} + S_{n-2} + S_{n-3} + \cdots + S_1 + 1 \\ S_{n-1} &= 2S_{n-2} + S_{n-3} + S_{n-4} + \cdots + S_1 + 1 \end{aligned}$$

Изваждаме второто от първото и получаваме

$$S_n - S_{n-1} = 2S_{n-1} + S_{n-2} - 2S_{n-2}$$

Всичко друго се съкращава. Това преписваме така

$$S_n = 3S_{n-1} - S_{n-2}$$

За да стане "истинско" рекурентно уравнение, трябва да му дадем и начални условия. Те са $S_1 = 1$ и $S_2 = 3$. И така,

$$S_n = \begin{cases} 1, & \text{ако } n = 1 \\ 3, & \text{ако } n = 2 \\ 3S_{n-1} - S_{n-2}, & \text{ако } n \geq 3 \end{cases} \tag{2}$$

Уравнение (2) е от втори ред и може да бъде решено с метода с характеристичното уравнение.

Ето решение на (2). Характеристичното уравнение е

$$x^2 - 3x + 1 = 0$$

Корените са $\frac{1}{2}(3+\sqrt{5})$ и $\frac{1}{2}(3-\sqrt{5})$. Оттук общото решение е

$$S_n = A\left(\frac{1}{2}(3+\sqrt{5})\right)^n + B\left(\frac{1}{2}(3-\sqrt{5})\right)^n$$

за някакви константи A и B. Да намерим A и B от началните условия.

$$S_1 = 1 = A\left(\frac{1}{2}(3+\sqrt{5})\right) + B\left(\frac{1}{2}(3-\sqrt{5})\right)$$
$$S_2 = 3 = A\left(\frac{1}{2}(3+\sqrt{5})\right)^2 + B\left(\frac{1}{2}(3-\sqrt{5})\right)^2$$

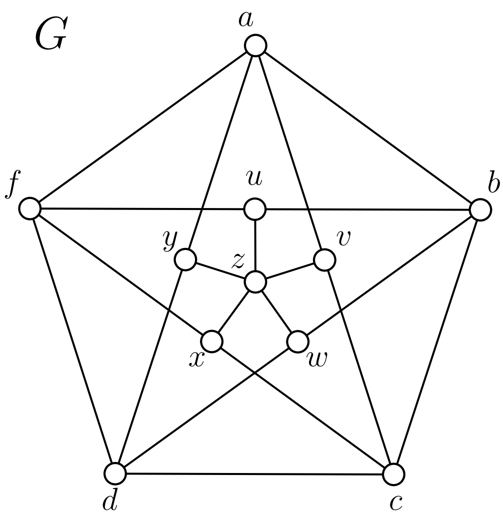
Оттук намираме $A = \frac{1}{\sqrt{5}}$, $B = -\frac{1}{\sqrt{5}}$, таке че

$$S_n = \frac{1}{\sqrt{5}}\left(\frac{3+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{3-\sqrt{5}}{2}\right)^n$$

Въпреки радикалите, S_n е цяло число за всяко $n \in \mathbb{N}^+$. Първите десет стойности са 1, 3, 8, 21, 55, 144, 377, 987, 2 584 и 6 765. Това са числа на Фибоначи, в което няма нищо странно, понеже $S_n = F_{2n}$ за всяко $n \in \mathbb{N}^+$, факт, който лесно може да се докаже по индукция.

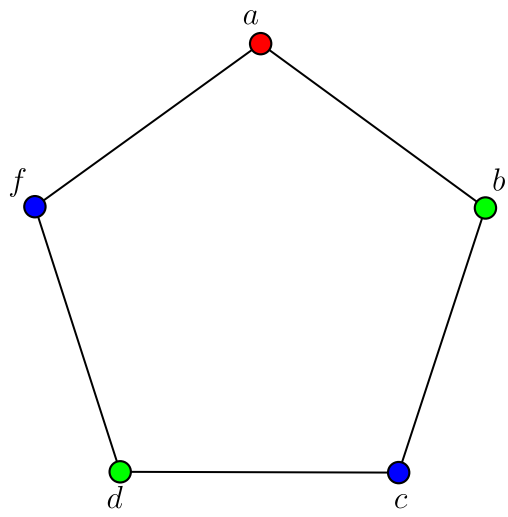
Задача 2: Припомнете си дефинициите на “кликово число на граф G”, което бележим с $\omega(G)$, и “хроматично число на граф G”, което бележим с $\chi(G)$. Намерете малък пример за граф G, такъв че $\omega(G) = 2$ и $\chi(G) = 4$. Нарисувайте графа ясно и прегледно и докажете формално и прецизно, че $\chi(G) = 4$. Това, че кликовото число е две, би трябвало да е очевидно от рисунката, но за хроматичното число нещата никога не са очевидни.

Решение: Разгледайте този граф G:



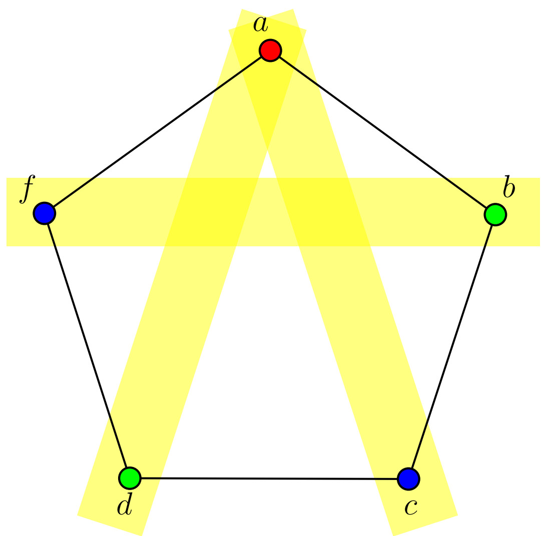
Твърдим, че $\omega(G) = 2$ и $\chi(G) = 4$. Това, че $\omega(G) = 2$, е очевидно от рисунката на графа: няма три върха, всеки две от които са съседи. Ще докажем, че $\chi(G) = 4$. Да допуснем, че $\chi(G) < 4$.

Да разгледаме цикъла a, b, c, d, f, a. Той е нечетен цикъл, следователно, съгласно изучаваното на лекции, не може да бъде оцветен с два цвята. Заключаваме, че $\chi(G) = 3$. БОО, нека цветовете са червен, зелен и син. Ето едно възможно оцветяване на цикъла в тези цветове:

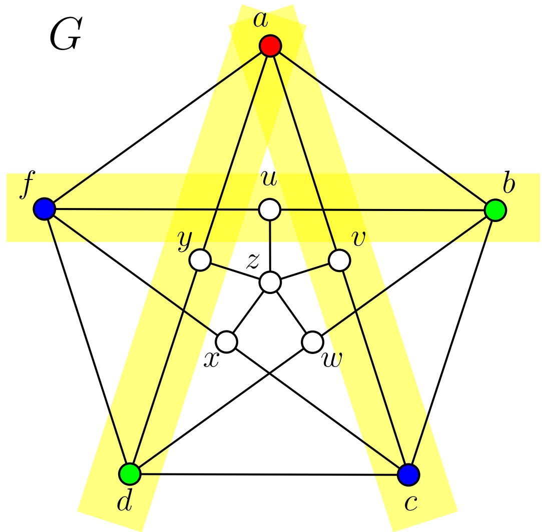


Възможни са и други оцветявания, но за всяко оцветяване в три цвята на този цикъл е вярно, че единият цвят се ползва точно веднъж, а всеки от други два се ползва точно два пъти – ако допуснем, че някой цвят се ползва поне три пъти, ще има ребро с двата края в един цвят, което не е разрешено. БОО, нека цветът, който се ползва точно веднъж, е червеният и нека a е червеният връх. Връх b е или зелен, или син. Ако b е зелен, единственото възможно оцветяване е това, което е показано. Ако b е син, оцветяването може да бъде довършено по единствен начин, а именно c е зелен, d е син и f е зелен. И така, щом цветът, който се ползва веднъж, е червеният и a е червен, върху останалите четири върха синият и зеленият цвят се редуват. БОО, разглеждаме само показаното оцветяване (b е зелен и т. н.).

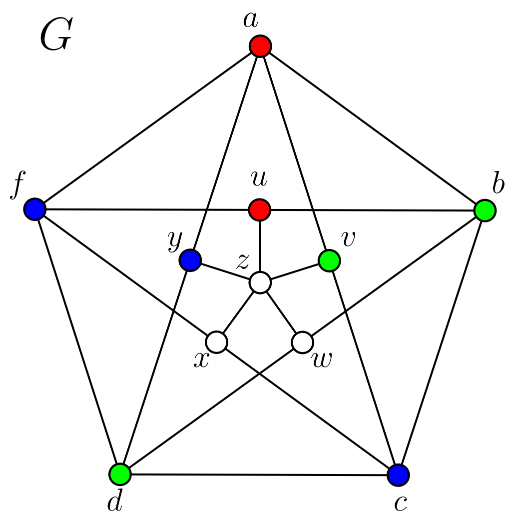
Има точно пет двойки върхове измежду a, $\ldots$, f, които не са съседи върху цикъла: $\{a, c\}$, $\{b, d\}$, $\{c, f\}$, $\{a, d\}$ и $\{b, f\}$. За три от тези двойки е вярно, че цветовете на двата върха в двойката са различни, а именно $\{a, c\}$, $\{a, d\}$ и $\{b, f\}$:



Да разгледаме избраното оцветяване на цикъла в контекста на целия граф:

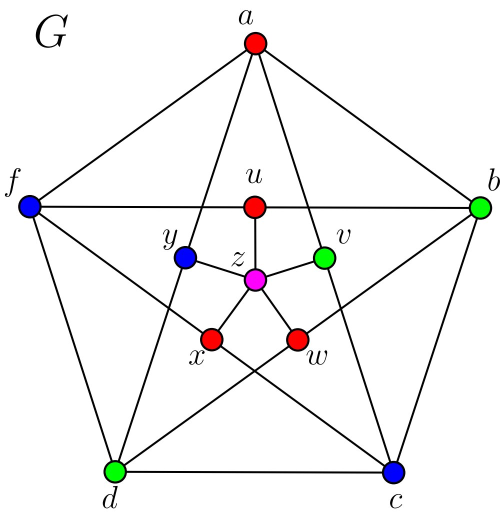


Щом ползваме само трите цвята червен, зелен и син, при това положение се налага y да е син, u да е червен и v да е зелен:



Но тогава z не може да е нито червен, нито зелен, нито син, бивайки съсед и на u, и на v, и на y. Полученото противоречие показва, че допускането $\chi(G) < 4$ е грешно.

От друга страна, G е 4-оцветим, както се вижда от следната рисунка, така че $\chi(G) = 4$.


G
a
b
c
d
f
u
v
w
x
y
z

Задача 3: Добре известно е, че всяко върхово оцветяване на граф в k цвята по същество е разбиване на множеството от върховете на k антиклики, като върховете от един цвят са една антиклика. Разгледайте следният алчен алгоритъм, който разбива множеството от върховете на антиклики.

Алгоритъм 1: Алгоритъм за разбиване на антиклики

Вход: граф $G = (V, E)$.
Изход: Разбиване на V на антиклики.

$\mathcal{U}$ е променлива от тип множество от множества от върхове.

❶ $\mathcal{U} \leftarrow \emptyset$.

❷ Намери антиклика $S \subseteq V$ с максимална мощност.

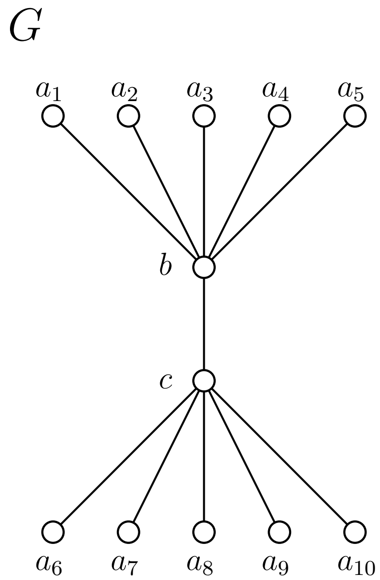
❸ Присвои $\mathcal{U} \leftarrow \mathcal{U} \cup \{S\}$.

❹ Изтрий върховете на S от G.

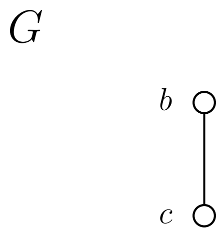
❺ Ако в G не са останали върхове, върни $\mathcal{U}$ и прекрати алгоритъма; в противен случай, иди на ред ❷.

Професор Дълбоков твърди, че този алгоритъм намира оптимално върхово оцветяване на входния граф. Опровергайте професора. Обосновете добре аргумента си.

Решение: За да опровергаем професора, достатъчно е да намерим един контрапример. Следният граф е контрапример:



На първата итерация алгоритъмът ще намери $S = \{a_1, a_2, \ldots, a_{10}\}$ на ред ❷, защото това е максималната антиклика в графа. След това $\mathcal{U}$ ще стане $\{\{a_1, a_2, \ldots, a_{10}\}\}$ на ред ❸. След това алгоритъмът ще изтрие върховете от $\{a_1, a_2, \ldots, a_{10}\}$ на ред ❹ и графът ще стане този:



Следва проверката на ред ❺. Тъй като все още има върхове, изпълнението ще отиде отново на ред ❷. В текущия граф максимална антиклика е всяко от $\{b\}$, $\{c\}$. Да кажем, че $S = \{b\}$. Тогава $\mathcal{U}$ ще стане $\{\{a_1, a_2, \ldots, a_{10}\}, \{b\}\}$, връх b ще бъде изтрит, цикълът ще се изпълни още веднъж и алгоритъмът ще върне $\mathcal{U} = \{\{a_1, a_2, \ldots, a_{10}\}, \{b\}, \{c\}\}$.

Но това разбиване на антиклики е върхово оцветяване в три цвята, понеже $|\mathcal{U}| = 3$. А всъщност входният граф е 2-оцветим, понеже е дърво; всяко дърво е 2-оцветимо, понеже е ацикличен граф, така че не съдържа нечетни цикли. Заключаваме, че посоченият алгоритъм не винаги намира оптимално върхово оцветяване.