ФП
01.VI.

Scheme:

- (if (> x 5)
  "iei"
  "oops")

- (let [(x 5) (y 3)] (+ x y)) ← създава се междинна среда, където x=5, y=3.

- (quote (1 2 3)) ≡ '(1 2 3)

Специални форми:
- define, lambda, if, cond
- let, let*, letrec, quote
- and, or
- delay, force ← за отложено оценяване

Апликативен | нормален метод
≡
Стриктно | мързеливо оценяване

↑ първо се оценяват аргументите на дадена ф-ия и после се извиква функцията с вече оценените аргументи

↑ обратното.

(f (+ 5 3)) →
(* (+ 5 3) (+ 5 3)) →
(* 8 (+ 5 3)) → (* 8 8)
→ 64

(define (f x) (* x x))

(f (+ 5 3)) → (f 8) → (* 8 8) → 64

- мързеливо, реално: (let [(x (+ 5 3))] (* x x))
→ (let [(x 8)] (* x x)) → (* 8 8) → 64

Оценяват се аргументите само веднъж, вместо експоненциално

```
(define (f x y)
  (if (> x 0)
      x
      (+ y 2)))
```

- мързеливо:

(f 5 (/ 1 0)) → (if (> 5 0)

5

(+ (/ 1 0) 2)) → 5

- стриктно:

(f 5 (/ 1 0)) → грешка при делене на 0

Полезен преговор за Haskell:

- Data.List
- pattern matching
- list comprehension

  [5 * x | x ← [1..10], even x]

- ламбда функции

  let f = \x → x*5

  let f = (* 5)

Полезен преговор за Scheme:

- map, filter, foldr, accumulate

Задачи

I)
21.03.09 / Инф / зад. 5

```
(define (count-sub l1 l2)
  (cond [(null? l2) 0]
        [(begins? l1 l2) (+ 1 (count-sub l1 (cdr l2)))]
        [else (count l1 (cdr l2))]))
```

```
(define (begins? l1 l2)
  (cond [(null? l1) #t]
        [(null? l2) #f]
        [(not (= (car l1) (car l2))) #f]
        [else (begins? (cdr l1) (cdr l2))]))
```

II. 10.09.2018 / Инф / задача 3.

```
(define (sumMaxRoots f ll)
  (apply +
    (foldr select list '()
      (map (lambda (l)
             (filter (lambda (x) (= (f x) 0)) l))
           ll))))
```

```
(define (select list l1 l2)
  (if (>= (length l1) (length l2)) l1 l2))
```

III

15.04.2014г | KH | зад. 7

```
magic f 0 = (\x -> x)
magic f 1 = f
magic f i
  | any (\x -> f1 x /= f2 x) [0..i] = f1 . f2
  | otherwise = totalMin [ magic f x | x <- [0..i-1] ]
     where f1 = magic f (i-1)
           f2 = magic f (i-2)

chainMinCompositions = [magic f i | i <- [0..]]
import Data.Ord (comparing)
import Data.List (minimumBy)
totalMin fs = minimumBy (comparing ($ 0)) fs

            (comparing (\f -> f 0))
```

– –

```
(define (chainMinComps f)
   (define (helper i)
      (cons-stream (magic f i) (helper (+ i 1))))
  (helper 0)
)
```

IV.

10.09.2009г /Инф/ зад. 3

```
(define (countMyHead lst)
  (cond [(null? lst) 0]
        [(null? (cdr lst)) 1]
        [(= (car lst) (cadr lst))
              (+ 1 (countMyHead (cdr lst)))]
        [else 1]
  )
)

(define (compress lst)
  (let [(count (countMyHead lst))]
    (if (count > 1)
        (cons (cons (car lst) count)
              (compress (drop count lst)))
        (cons (car lst) (compress (cdr lst))))))
```

При потоци: решаване за списък и после променяме дребни работи за работа с потоци вместо списъци

```
perm [] = [[]]
perm lst = [(x:p) | x <- lst, p <- perms (delete x lst)]
```

Графи

DC
02.06

- $d(v)$ = степен на връх v = брой ребра, инцидентни с v (неориентиран граф)
- $G = (V, E)$, $|V| = n$, $|E| = m$, $0 \le m \le \binom{n}{2} = \frac{n(n-1)}{2}$

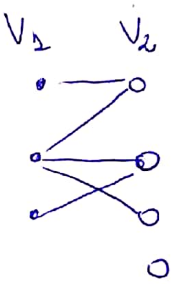
↑ за неориентиран граф без примки

$m \le \frac{n(n-1)}{2} + n$ ← с примки

- $\sum_{v} d(v) = 2m$ // върховете от нечетна степен са четен брой

- Двуделен граф

$$\sum_{v \in V_1} d(v) = m = \sum_{v \in V_2} d(v)$$

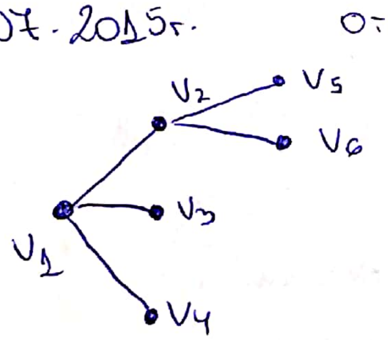


- Кликово число $\omega(G)$ - брой върхове в най-голямата клика
- Хроматично число $\chi(G)$ - минимален брой цветове, нужни за оцветяване на върховете, така че всяко ребро да има за краища върхове с различен цвят
- Хроматичен индекс - същото, но за ребра

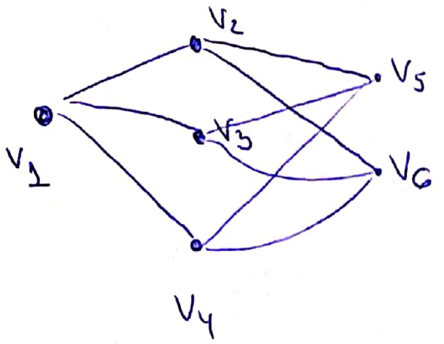
Задачи

1) 14.07.2015г.

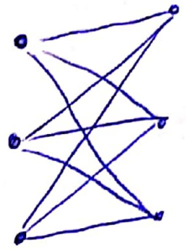
а)



От v_2 не може ребра към v_1, v_3 и v_4, защото няма цикли по условие => добавяме нови => станаха поне 6 б)



или



Th: В двуделен граф няма цикъл с нечетна дължина

// K_n - пълен граф

Отг: Да, има: $K_{3,3}$

---

2) 11.07.2017

Разглеждаме неориентиран граф, такъв че

- върховете: числата $0, 1, \ldots, n-1$
- ребрата: плочките

Предполагаме, че комплектът домино е пълен
=> Графът е пълен и на всеки връх има примки.
((3,3); (9,9))

Ойлеров път - през всяко ребро
Хамилтонов път - през всеки връх

В случая търсим Ойлеров цикъл:
Всяка затворена верига от плочки е Ойлеров цикъл и обратното.

Th: G има Ойлеров цикъл <=> G е свързан и всеки връх е от четна степен

Th: G съдържа Ойлеров път, който не е цикъл $\Leftrightarrow$ G е свързан и има точно 2 върха от нечетна степен.

• Всяка примка се брои 2 пъти към степента на връх

$$\forall v: d(v) = (n-1) + \underbrace{2}_{\text{примка}} = n+1$$

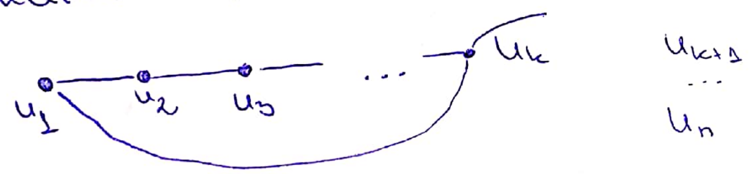
• Графът е пълен => Графът е свързан.

• Кога $d(v)$ е четно? $d(v) = n+1$ => n трябва да е нечетно. □

---

3) 09.09.2017г.

Допускаме противното, че u_1 и u_k са съседи.



Има поне още един връх освен $u_1 \dots u_k$, защото $k < n$ по условие.

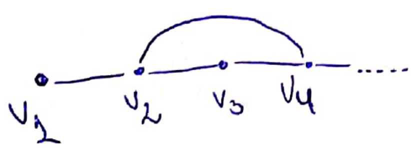
Б.О.О считаме, че ребро излиза от u_k към другия връх, защото графът е свързан.

Б.О.О. считаме, че u_{k+1} се свързва с u_k. Получихме по-дълъг път от $u_1 \dots u_k$, което е противоречие. □.

---

4) 15.07.2014г.

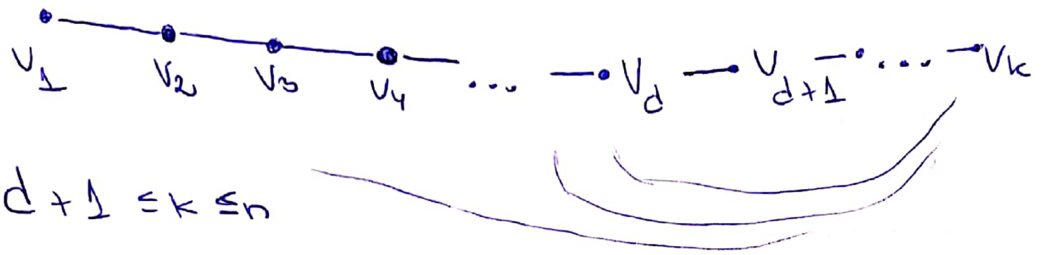
a)



Графът не е безкраен => на n-та стъпка ще трябва да повторим връх

// Граф без цикли е гора. = множество от дървета
Ако графът е свързан => дърво

б)



Няма повече нови върхове

$d+1 \le k \le n$

Нека $v_k = u_0$

$v_{k-1} = u_1$ .... $-u_{d+1} - u_d - \dots - u_3 - u_2 - u_1 - u_0$

⋮

Нека $u_0 \dots u_k$ е най-дълъг път в G.

От u_0 излизат поне d ребра.

От u_0 не може да излиза ребро към връх, различен от $u_1 \dots u_k$, защото ще е противоречие с най-дълъг път.

Тогава $d \le k$.

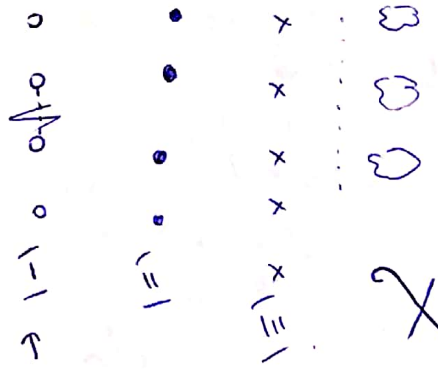
d-тото ребро: $(u_0; u_p)$ $p \ge d$ е цикъл с дължина поне $d+1$

$u_0 - u_1 - u_2 - \dots - u_p$

□

5) 13.07.2018г

Обикновен граф = неориентиран, ненулти, без примки



$\chi(G)$

↑ няма ребра м/у върхове от 1 цвят

тривиално...

Важното е, че с $\chi(G)-1$ цвята не може да се оцвети графа, а не, че с $\chi(G)$ може

$$\binom{\chi(G)}{2} \le m$$

← защото между всеки 2 цвята има ребро

Не може да няма нито 1 ребро между 2 колонки

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!} = \frac{n.(n-1).(n-2)\ldots(n-k+1)}{k!}$$

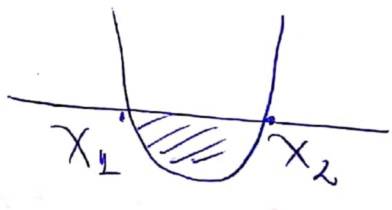
$$\binom{n}{2} = \frac{n(n-1)}{2}$$

$$\binom{\chi(G)}{2} = \frac{\chi(G)(\chi(G)-1)}{2} \le m$$

$$\chi(G)^2 - \chi(G) - 2m \le 0$$

$$\chi(G)_{1,2} = \frac{1 \pm \sqrt{1+8m}}{2} = \frac{1}{2} \pm \sqrt{\frac{1}{4}+2m}$$

$$x_1 \le \chi(G) \le x_2$$



$$\Rightarrow \chi(G) \le \frac{1}{2} + \sqrt{\frac{1}{4}+2m}$$

□

6) 10.09.2018г.

Допускаме противното:

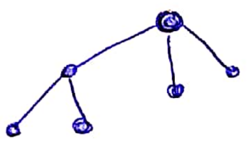
$\exists v_0 \in V : d(v_0) \le k-2.$

$\chi(G-v_0) = k-1$ $\Big\}$ $\chi(G) = k-1 \Rightarrow$ ↯

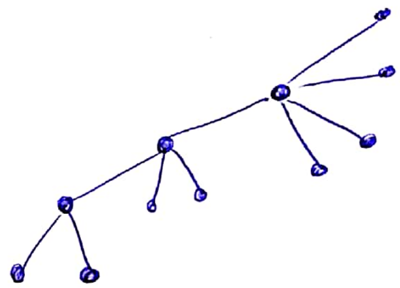
↑ защото $d(v_0) \le k-2$ и няма да има ребро с поне един от цветовете и k-ия цвят няма да ни трябва. □

7) 05.07.2008г

T_2:



T_3:



Нека a_n = брой вътрешни върхове на T_n.

$a_n = ?$

$a_0 = 1$

$a_1 = 1$

$a_2 = 2$

$a_3 = 3$

$a_n = a_{n-1} + 2a_{n-2} - 1, n^0 \cdot 1^n$

$x^n = x^{n-1} + 2x^{n-2} \quad // : x^{n-2}$

$x^2 = x + 2$

$x^2 - x - 2 = 0$

Scanned by CamScanner

$$x_{1,2} = \frac{1 \pm \sqrt{9}}{2} = \frac{1 \pm 3}{2}$$

$x_1 = 2$ $\quad x_2 = -1$

$\Rightarrow \{-1; 2\}_M$

От свободния член $\Rightarrow \{1\}_M$

$\Rightarrow \{-1; 2\} \cup \{1\}_M = \{-1; 1; 2\}_M$

$\Rightarrow a_n = c_1 . (-1)^n + c_2 . 1^n + c_3 . 2^n$

$$\begin{cases} c_1 + c_2 + c_3 = 1 \\ -c_1 + c_2 + 2c_3 = 1 \\ c_1 + c_2 + 4c_3 = 3 \end{cases}$$

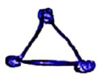
$c_1 = \frac{1}{6}$

$c_2 = \frac{1}{2}$

$c_3 = \frac{1}{3}$

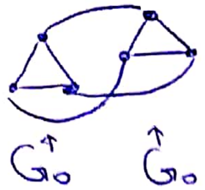
Отг: $a_n = \dfrac{(-1)^n + 3 + 2^{n+1}}{6}$

□

8) $(G_n)_{n=0}^{\infty}$ - редица от графи: $G_0 = K_3$: 

G_{n+1} е образувано от 2 копия на G_n, като ребрата между копията свързват съответните върхове на двете копия, т.е v_1 с v_1, v_2 с v_2, ...

G_1:



$G_n = (V_n, E_n)$

Да се намерят броя на върховете и броят на ребрата на $n^{ия}$ граф.

G_{n+1} = две копия на G_n $\Rightarrow |V_{n+1}| = 2|V_n|$

$|V_0| = 3$

$\boxed{\text{Отг: } |V_n| = 3.2^n}$

За ребрата:

$$|E_{n+1}| = 2.|E_n| + |V_n|$$
$$= 2|E_n| + 3.2^n.n^0$$

$$x^{n+1} = 2x^n$$
$$x - 2 = 0$$
$$x = 2 \quad \Rightarrow \{2\}_M$$

O: $3.2^n.n^0 \Rightarrow \{2\}_M$

$\Rightarrow \{2\}_M \cup \{2\}_M \Rightarrow \{2, 2\}_M$

$\Rightarrow |E_n| = (C_1 + C_2 n).2^n$

$|E_0| = 3$

$|E_1| = 9$

$(c_1 + c_2 \cdot 0)\cdot 2^0 = 3 \quad \Rightarrow c_1 = 3$

$(c_1 + c_2)\cdot 2 = 9 \quad \Rightarrow 3 + c_2 = \frac{9}{2}$

$c_2 = \frac{9}{2} - \frac{6}{2} = \frac{3}{2}$

$|E_n| = \left(3 + \frac{3n}{2}\right)\cdot 2^n = 3(2+n)\cdot 2^{n-1}$ □

---

9) 11.09.2014

б) $\sum_{v\in V_1} d(v) = \sum_{v \in V_2} d(v)$

$|V_1|\cdot k = k|V_2|$

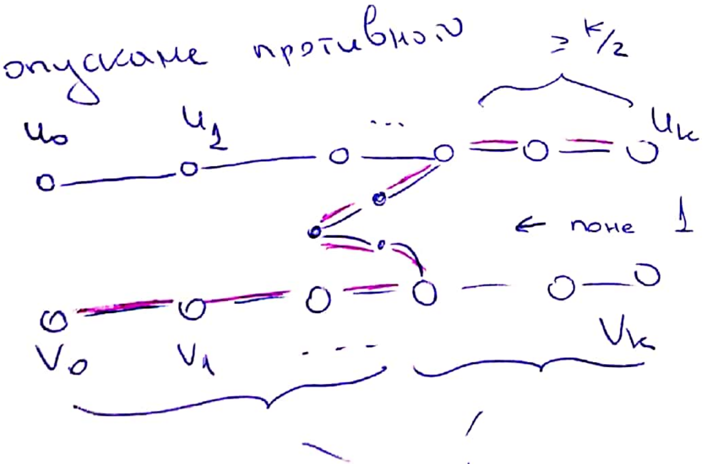
$\Rightarrow |V_1| = |V_2|$

---

10) 10.09.2015г.

Най-дългите прости пътища са 6 у.

а) Допускаме противното



поне 1 част има дължина $\geq \frac{k}{2}$

това важи и за горната част.

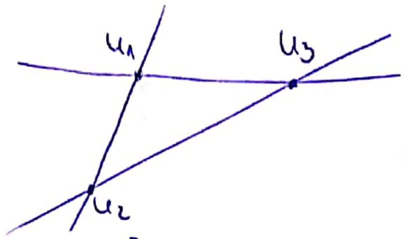
$\Rightarrow$ има друг път с дължина $\geq \frac{k}{2} + 1 + \frac{k}{2} = k+1 \Rightarrow$ ↯

дължината на най-дългия път беше k.

Б) Задачата е интересна за $|Y| \geq 3$.

Допускаме противното.

Тогава



- по допускане

Ако $u_1 = u_2 \to$ цикъл

Б.О.О

$\Rightarrow u_1 \neq u_2 \neq u_3 \neq u_1$

Но и в този случай пак има цикъл $\Rightarrow$ ↯