1. V jaké třídě složitosti je funkce X v následujícím programu vzhledem k velikosti pole p (budeme ji značit n a odpovídá hodnotě proměnné N)?

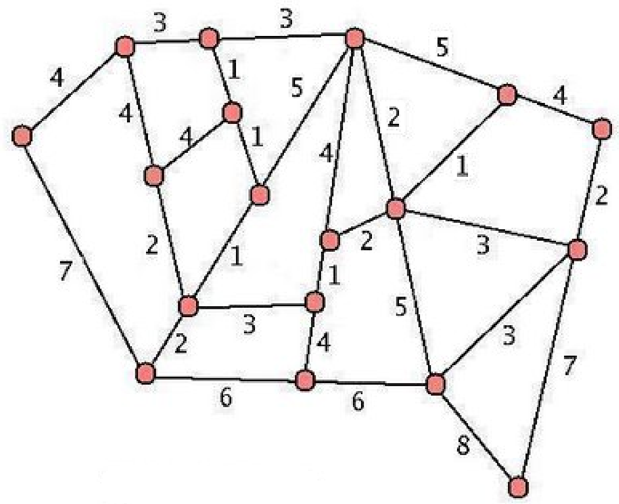
```
int N;
int * p;

void X (void)
{   int i,j,t;
    for (i = N; i > 0; i--) {
        t = 0;
        for(j = 1; j < i; j++)
            if (p[j-1] > p[j]) {
                p[j]   = p[j-1] - p[j];
                p[j-1] = p[j-1] - p[j];
                p[j]   = p[j-1] + p[j];
                t = 1;
            }
        if (t == 0) break;
    }
}
```

   a) O(1)
   b) O(n*log(n))
   c) <u>O(n^2)</u>
   d) <u>Ω(1)</u>
   e) <u>Ω(log(n))</u>
   f) Ω(n^2)
   g) Θ(n^2)
   h) Θ(1)

2. V jaké amortizované třídě složitosti je funkce X z předchozího programu vzhledem k velikosti pole p (budeme ji značit n a odpovídá hodnotě proměnné N)?

   a) O(1)
   b) <u>O(n*log(n))</u>
   c) <u>O(n^2)</u>
   d) <u>Ω(1)</u>
   e) <u>Ω(log(n))</u>
   f) Ω(n^2)
   g) Θ(n^2)
   h) Θ(1)

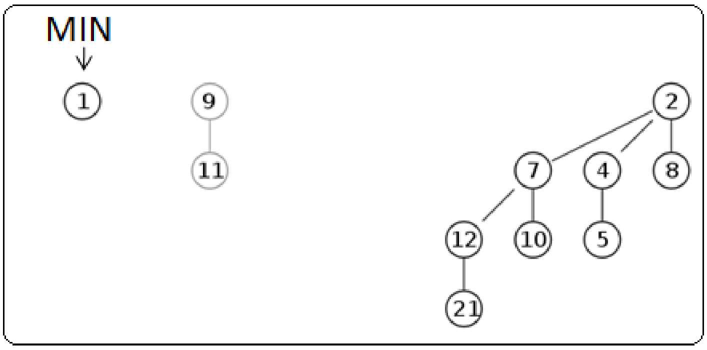
3. Jaká je celková váha minimální kostry následujícího grafu?

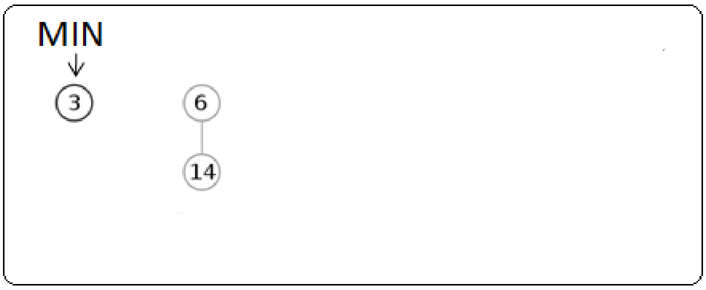


   a) 30
   b) 37
   c) <u>42</u>
   d) 45
   e) 50

4. Jaký nejmenší počet hran má graf se čtyřmi komponentami souvislosti na 9987 vrcholech?
   a) 9986
   b) <u>9983</u>
   c) 9980
   d)

5. Mějme následující dvě binomiální haldy:





Kolik stromů bude obsahovat výsledná halda, která vznikne operací merge?

a) 1
b) 2
c) <u>3</u>
d) 4

(číslování otázek v tomto dokumentu je nepodstatné)

12x.

Je dán zásobníkový automat P s jediným stavem q, který přijímá slova prázdným zásobníkem.Vnější abeceda P je $\{a, b\}$, zásobníková abeceda P je $\{S, A, B\}$. Na začátku práce je v zásobníku symbol S. Přechody P jsou dány vztahy (vrchol zásobníku je vlevo):

$\delta(q, a, S) = (q, ABS)$
$\delta(q, a, A) = (q, \varepsilon)$
$\delta(q, a, B) = (q, B)$
$\delta(q, b, B) = (q, \varepsilon)$
$\delta(q, b, B) = (q, BB)$
$\delta(q, b, S) = (q, B)$

Poté, co automat přečte první čtyři znaky slova *aabbaab* a provede příslušné přechody, je na zásobníku řetězec (vrchol zásobníku je vlevo).

a) ***B***
b) ***BS***
c) *BBS*
d) *BBB*
e) ***BBBS***

Automat je nedeterministický, možností, co provede, je proto více. Např.
a) $(q, aabbaa, S) \rightarrow (q, abbaa, ABS) \rightarrow (q, bbaa, BS) \rightarrow (q, baa, S) \rightarrow (q, aa, B)$ nebo
b) $(q, aabbaa, S) \rightarrow (q, abbaa, ABS) \rightarrow (q, bbaa, BS) \rightarrow (q, baa, BBS) \rightarrow (q, aa, BS)$ nebo
e) $(q, aabbaa, S) \rightarrow (q, abbaa, ABS) \rightarrow (q, bbaa, BS) \rightarrow (q, baa, BBS) \rightarrow (q, aa, BBBS)$.
A podobně...

Varianta: Která slova z dané množiny slov automat přijme?
Nebo: Po přečtení 3 znaků poznejte podle stavu zásobníku, které znaky to byly.)

13x.

Jsou dány dvě gramatiky G_1 a G_2. Označme $L(G_1)$ resp. $L(G_2)$ jazyk generovaný gramatikou G_1 resp. G_2. Zaškrtněte platná následující tvrzení.

G_1:
$S \rightarrow ABcd$
$A \rightarrow aAa \mid aa$
$B \rightarrow bB \mid b$

G_2:
$S \rightarrow Ed$
$E \rightarrow Fc$
$F \rightarrow Db$
$D \rightarrow Db$
$D \rightarrow Ca$
$C \rightarrow Ba$
$B \rightarrow Ca$
$B \rightarrow \varepsilon$

a) $\boldsymbol{L(G_1)}$ **je bezkontextový jazyk.**
b) $\boldsymbol{L(G_2)}$ **je bezkontextový jazyk.**
c) $\boldsymbol{L(G_1)}$ **je regulární jazyk.**

d) $L(G_2)$ je regulární jazyk.
e) $L(G_1) = L(G_2)$.

V tomto případě platí všechna tvrzení.
Je nutno rozlišovat mezi bezkontextovým jazykem a bezkontextovou gramatikou. Regularní jazyk lze generovat gramatikou, která je bezkontextová a není regulární.

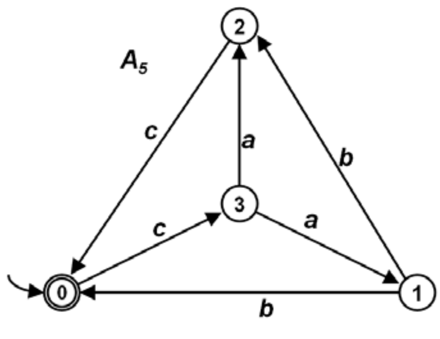
(Varianta: Určete, zda daná slova patří do zřetězení, iterace, doplňku daných jazyků, příp. jejich sjednocení, průniku, etc.)

14x.
Jsou dány dva jazyky R_1 a R_2 nad abecedou $A = \{0, 1, 2\}$. R_1 obsahuje právě všechna taková slova z A^+, která neobsahují podřetězce 10, 20, 21. R_2 obsahuje všechna slova $w \in A^+$, pro něž platí: $w_1 = \in A$, $w_n \in A$, $w_i = 1$ $(1 < i < n)$, $n = |w| \geq 2$.
Určete, která z následujících tvrzení platí (tečka mezi identifikátory jazyků představuje operátor zřetězení jazyků).

a) R_1 je regulární jazyk.
b) R_2 je zároveň bezkontextový i regulární jazyk.
c) $01020 \in R_1 \cap R_2$.
d) $01000 \in R_1.R_2$.
e) $00000 \in R_1.R_2$.

(Varianta: Určete gramatiku daných jazyků, jejich mohutnost, mohutnost průniku. Určete, zda sjednocení nebo průnik nebo zřetězení nebo doplněk jazyků je ve vztahu inkluze vůči sjednocení nebo průnik nebo zřetězení nebo doplňku jazyků, apod.)

15x.
Je dán NKA A_5 pomocí přechodového diagramu na obrázku. Po provedení standardního algoritmu převodu NKA na DKA vznikne DKA A_{51}, pro nějž platí (přechodovou funkci v A_{51} označme δ):



a) A_{51} má tři koncové stavy.
b) A_{51} má celkem pět stavů.
c) Každý stav A_{51} odpovídá jednomu nebo dvěma stavům A_5.
d) $\delta(02, c) = 03$.
e) $\delta(03, a) = 12$.

V tomto případě platí všechna tvrzení, tabulka přechodů A_{51} je vpravo

	a	b	c	
0			**3**	**F**
3	**12**			
12		**02**	**0**	
02			**03**	**F**
03	**12**		**3**	**F**

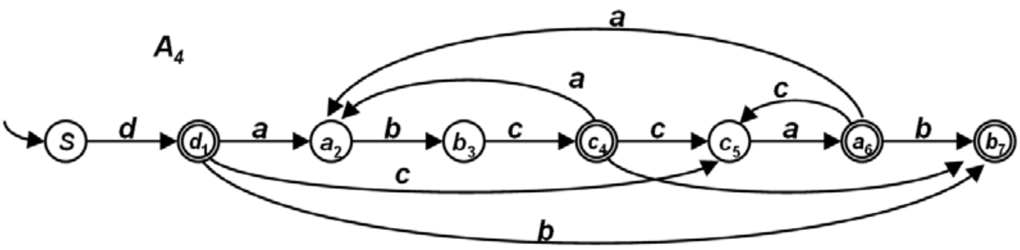
(Varianta: Nic moc, vždy jde asi o přímočaré nasazení algoritmu převodu NKA na DKA.)

16x.

Nedeterministický automat A_4 vznikl aplikací algoritmu pro tvorbu automatu vyhledávajícího v textu každý vzorek, který vyhovuje regulárnímu výrazu R = $d(abc+ca)^*b^*$. Označme Platí:

a) A_4 má 8 stavů.
b) A_4 má 3 koncové stavy.
c) Přechodový diagram A_4 je orientovaný graf, jehož vrcholy jsou stavy A_4, hrany jsou přechody A_4 a tento graf je acyklický.
d) V A_4 lze po přečtení jednoho znaku přejít ze startovního stavu do koncového stavu.
e) Počáteční stav A_4 není koncovým stavem.

A_4 vidíme na obrázku.



(Varianta: Vyhledávání řetězce s určitou vzdáleností od vzorku, vyhledávání libovolného prvku z dané konečné množiny vzorků, vyhledávání libovolného podřetězce daného vzorku)

17x.

Gramatika G_3 má neterminální symboly A, B, C, D, E, S, terminální symboly a, b, c, d, e, f, startovní symbol S a pravidla:

$S \rightarrow AE$
$A \rightarrow ED$
$A \rightarrow BC$
$B \rightarrow cD$
$B \rightarrow eE$
$B \rightarrow d$
$C \rightarrow bCa$
$C \rightarrow d$
$D \rightarrow aD$
$E \rightarrow fE$
$E \rightarrow \varepsilon$

Po rozšíření množiny pravidel o pravidlo P přestane být G_3 $LL(1)$ gramatikou.

a) $P = D \rightarrow ab$
b) $P = S \rightarrow aB$
c) $P = C \rightarrow eA$
d) $P = B \rightarrow \varepsilon$
e) $P = E \rightarrow d$

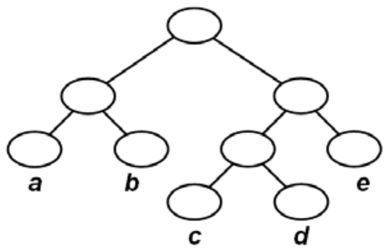
$D \rightarrow ab$ způsobí konflikt: $a \in$ FIRST(ab) a zároveň $a \in$ FIRST(aD).
$S \rightarrow aB$ způsobí konflikt: $a \in$ FIRST(aB) a zároveň $a \in$ FIRST(AE).

$B \rightarrow \varepsilon$ způsobí konflikt: $d \in$ FIRST(d) a zároveň $d \in$ FOLLOW(B).
$E \rightarrow d$ způsobí konflikt: $d \in$ FIRST (ED) a zároveň $d \in$ FIRST(BC).

(Varianta: Množina FIRST(něco), FOLLOW(něco jiného) pro danou gramatiku.)

18x.

Statický Huffmanův strom kódované zprávy má tvar uvedený na obrázku. Možné frekvence jednotlivých znaků tedy jsou:



a) ***a*: 10, *b*: 10, *c*: 10, *d*: 10, *e*: 10**
b) *a*: 10, *b*: 10, *c*: 20, *d*: 20, *e*: 10
c) *a*: 10, *b*: 10, *c*: 10, *d*: 30, *e*: 10
d) *a*: 10, *b*: 30, *c*: 10, *d*: 10, *e*: 30
e) ***a*: 20, *b*: 20, *c*: 10, *d*: 10, *e*: 40**

(Varianta: Adaptivní strom pro danou zprávu.)

Zde je nutno některé varianty vyloučit rovnou, jako např. b), kde mají znaky s nejdelším kódem *c* a *d* zároveň největší frekvenci, což pro optimalitu komprese je holý nesmysl.
V jiných případech může být nutno postavit strom, např. varianta e) je kritická, po zvýšení frekvence znaku *e* na 41 již daný strom vyhovovat nebude.

19x.

V textu se vyhledává vzorek *P* pomocí Boyer–Moore algoritmu. Text i vzorek jsou nad abecedou{*a, b, c, d*}. Víme, že Tabulka Bad Character Shift pro vzorek *P* obsahuje (kromě jiných hodnot) hodnotu 3 a 5. Určete, která z nabízených variant určitě NENÍ vzorkem *P*.

a) ***aaaaa***
b) ***bbbb***
c) ***ccc***
d) ***ddddbb***
e) ***abcdaaaaad***

V tomto případě platí všechna tvrzení.

Definice obsahu oné tabulky by měla stačit k vyřešení úlohy tothoto druhu.

(Varianta: Good suffix shift.)

20x.

Gramatika G_2 má neterminální symboly A, B, C, D, S, terminální symboly a, b, c, d, e, startovní symbol S a pravidla:

(1) $S \rightarrow aAbD$
(2) $A \rightarrow \varepsilon$
(3) $A \rightarrow d$
(4) $A \rightarrow BC$
(5) $B \rightarrow cCD$
(6) $B \rightarrow \varepsilon$
(7) $C \rightarrow aDeCd$
(8) $D \rightarrow eA$

Označme $M[X][y]$, resp $M[X][\varepsilon]$ prvek rozkladové tabulky gramatiky G_2, který odpovídá neterminálu X a terminálu y, resp. neterminálu X a prázdnému řetězci ε. Určete, která tvrzení o tabulce M platí.

a) $\boldsymbol{M[S][d]}$ **není definováno (= chyba v analyzovaném slově)**
b) $\boldsymbol{M[A][\varepsilon] = \varepsilon, 2}$
c) $\boldsymbol{M[B][a] = \varepsilon, 6}$
d) $\boldsymbol{M[C][c]}$ **= není definováno (= chyba v analyzovaném slově)**
e) $\boldsymbol{M[D][e] = eA,}$ **8.**

Pravidlo A $\rightarrow \varepsilon$ se dostane do ε sloupce tabulky, pokud zároveň $\varepsilon \in$ FOLLOW(A), což v případě G_2 platí díky pravidlům (1) $S \rightarrow aAbD$ a (8) $D \rightarrow eA$. Proto platí b). Neplatnost a) a d) je jasná z definice tabulky. Pravidlo (6) $B \rightarrow \varepsilon$ použijeme k expanzi zásobníku, když čtený symbol je *a,* protože jediný možný další symbol na zásobníku je C (jen jediné pravidlo $A \rightarrow BC$ má symbol B na pravé straně) a ten zase lze expandovat jedině na $aDeCd$. Odtud plyne $M[B][a] = \varepsilon$, 6.Ostatní případy řešíme obdobně.

(Varianta: Jaká je posloupnost (nebo její část) čísel pravidel v levém rozkladu vytvořeném při analýze konkrétního slova v konkrétní gramatice. Ze zadané tabulky určit některé vlastnosti gramatiky.)