

КОНТРОЛНО № 2 ПО ИЗБИРАЕМАТА УЧЕБНА ДИСЦИПЛИНА  
 “ДИЗАЙН И АНАЛИЗ НА АЛГОРИТМИ (ВТОРА ЧАСТ)”,  
 20 ЯНУАРИ 2022 Г. — СУ, ФМИ

**Задача 1.** Какво извършва следният програмен код, написан на езика Си-плюс-плюс, и каква е амортизираната му времева сложност? Дайте кратък отговор без подробна обосновка.

```
bool nextObj(std::vector<size_t>& v, size_t n)
{
    int j = v.size() - 1;

    while (j >= 0 && v[j] == n - v.size() + j + 1)
        j--;

    if (j < 0)
        return false;

    v[j]++;

    for (size_t i = j + 1; i < v.size(); i++)
        v[i] = v[i - 1] + 1;

    return true;
}
```


**Задача 2.** Изчислете разстоянието на Левенщайн между думите СЕНО и ЯСЕН, попълвайки таблицата. Оградете с кръгче числото в клетката, от която се намира отговорът, т.е. търсеното разстояние между двете думи. Оцветете (но така, че да се четат числата) онези клетки, по които се търси най-късото редакционно предписание. Словесно опишете най-късото редакционно предписание, чрез което от думата СЕНО се получава думата ЯСЕН.

							Я	С	Е	Н
С										
Е										
Н										
О										

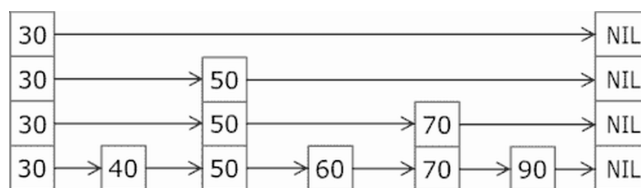
**Задача 3.** Декодирайте данните, получени след компресиране с алгоритъма LZ77: (0; 0; А), (0; 0; В), (0; 0; Р), (3; 1; К), (2; 1; Д), (7; 4; Ψ).

**Задача 4.** Чрез алгоритъма на Гейл—Шепли намерете система от устойчиви бракове за дадените на картинката предпочитания, ако мъжете предлагат брак на жените. Когато няколко мъже са свободни едновременно, прави предложение онзи от тях, който е първи по име в азбучния ред.

Подробно опишете стъпките и отговора на алгоритъма.

 Антон	 Дора, Елена, Галя	 Галя	 Борис, Антон, Васил
 Борис	 Галя, Дора, Елена	 Дора	 Борис, Васил, Антон
 Васил	 Галя, Дора, Елена	 Елена	 Антон, Борис, Васил

**Задача 5.** Покажете пътя, по който числото 90 се търси в дадения списък с пропуски.



**Задача 6.** Постройте минималния автомат на наставките за знаковия низ *aba*.

**Задача 7.** Постройте и попълнете сегментно дърво, с което да се смятат бързо сборовете от елементите на произволни подмасиви на числовия масив  $A = (5; 3; 10; 4; 1; 9; 6; 7)$ . Обяснете как се използва дървото за намиране на сбора от елементите на оградения подмасив; към словесните обяснения добавете чертеж, на който са отбелязани върховете на дървото, чиито стойности трябва да се съберат (тези върхове не са непременно листа).

**Задача 8.** Следният програмен код на Си-плюс-плюс е част от многонишкова програма, чието конкретно предназначение не ни интересува.

```

63         while (true)
64         {
65             int input;
66             cin >> input;
67             if (input >= 0)
68             {
69                 std::lock_guard<std::mutex> guard(workProtect);
70                 tasks.push(input);
71             }
72             else
73                 break;
74             cv.notify_one();
75         }
76         {
77             std::lock_guard<std::mutex> guard(workProtect);
78             isWorking = false;
79         }
80         cv.notify_one();
81         th.join();
82         std::cout << "Done!" << std::endl;

```

Двойката фигурни скоби на редовете № 64 и № 75 от кода огражда блок от оператори, който е тялото на цикъла с предусловие от ред № 63. Двойката скоби на редовете № 68 и № 71 също огражда блок от оператори, който е тялото на условния оператор от ред № 67.

За какво служат обаче скобите от редовете № 76 и № 79? Те също създават блок от оператори (тези от редовете № 77 и № 78), но защо изобщо е нужен блок на това място, щом като тук няма нито цикъл, нито условен оператор, нито оператор за избор на вариант?

**Време за работа:** два астрономически часа (т.е. сто и дващест минути).

**Оценката** по шестобалната система е равна на броя на задачите, решени вярно и пълно. Не се зачитат решения, които не удовлетворяват всички изисквания, поставени в условията. Използването на българска терминология е задължително!

## РЕШЕНИЯ

**Задача 1.** Програмният код поражда комбинациите на целите числа от 1 до  $n$  вкл. без повторение, от клас  $k = v.size()$ . По-точно, по дадена комбинация програмният код поражда следващата комбинация в лексикографската наредба на комбинациите. Може да се докаже, че амортизираната времева сложност е  $\Theta(1)$  при  $k \leq \frac{n}{2}$ . При  $k > \frac{n}{2}$  е по-добре да генерираме допълненията на комбинациите от клас  $k$ , тоест комбинациите от клас  $n - k$ .

**Задача 2.** От попълнената таблица става ясно, че разстоянието на Левенщайн между думите СЕНО и ЯСЕН е равно на 2 (ограденото число в долния десен ъгъл). Клетките, боядисани с жълт цвят, образуват маршрута, по който се намира най-късото редакционно предписание, чрез което от думата СЕНО се получава думата ЯСЕН. То се състои от две стъпки:

	Я	С	Е	Н	
0	0	1	2	3	4
С	1	1	1	2	3
Е	2	2	2	1	2
Н	3	3	3	2	1
О	4	4	4	3	2

- 1) В началото на думата СЕНО добавяме буквата Я и получаваме знаковия низ ЯСЕНО.
- 2) От края на редицата ЯСЕНО изтриваме буквата О и получаваме думата ЯСЕН.

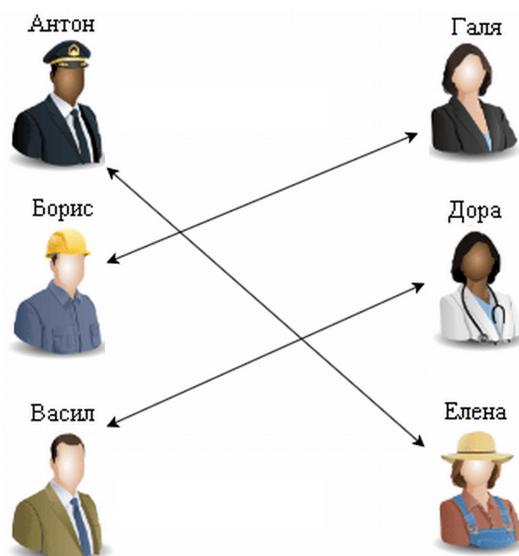
**Задача 3.** След декодиране се получава знаковият низ АБРАКАДАБРАΨ.

**Задача 4.** Прилагаме алгоритъма на Гейл—Шепли стъпка по стъпка:

- 1) Отначало всички мъже и жени са свободни.
- 2) Антон предлага брак на Дора. Дора приема предложението. Антон и Дора се сгодяват.
- 3) Борис предлага брак на Галя. Галя приема предложението. Борис и Галя се сгодяват.
- 4) Васил предлага брак на Галя. Галя отхвърля предложението, защото предпочита Борис.
- 5) Васил предлага брак на Дора. Дора приема предложението, защото предпочита Васил. Дора се разделя с Антон и се сгодява за Васил.
- 6) Антон предлага брак на Елена. Елена приема предложението. Антон и Елена се сгодяват.
- 7) Всички са сгодени, тоест няма свободни мъже и жени. Годениците сключват бракове.

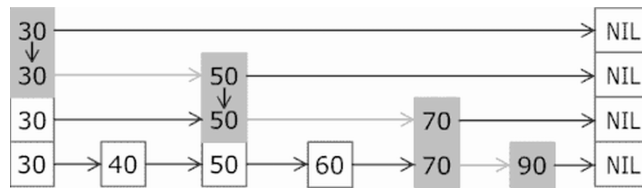
Резултатът от работата на алгоритъма е съвкупност от три брачни двойки:

Антон и Елена; Борис и Галя; Васил и Дора.



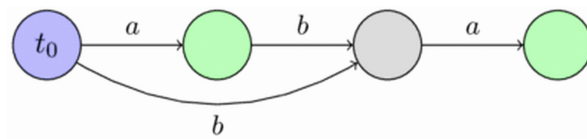
Изображенията са от <https://www.arvarik.com/visiting-the-stable-marriage-problem> (страница в Интернет, посветена на задачата за устойчивите бракове).

**Задача 5.** Пътят, по който числото 90 се търси в дадения списък с пропуски, е показан на чертежа.



Картинките в условието и решението на тази задача са от статията *Списък с пропуски* от *Википедия*.

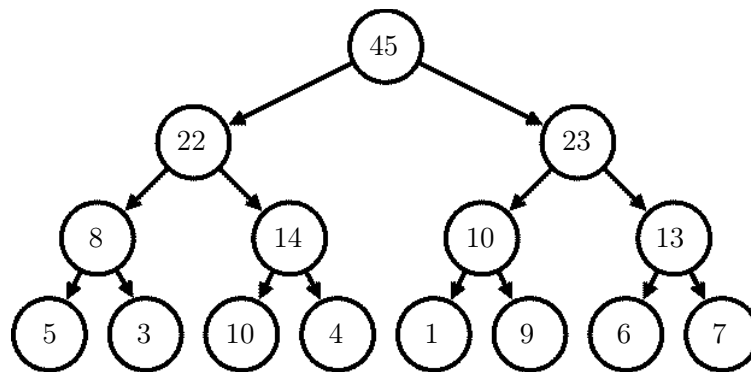
**Задача 6.** Минималният автомат на наставките за знаковия низ *aba* е показан на чертежа.



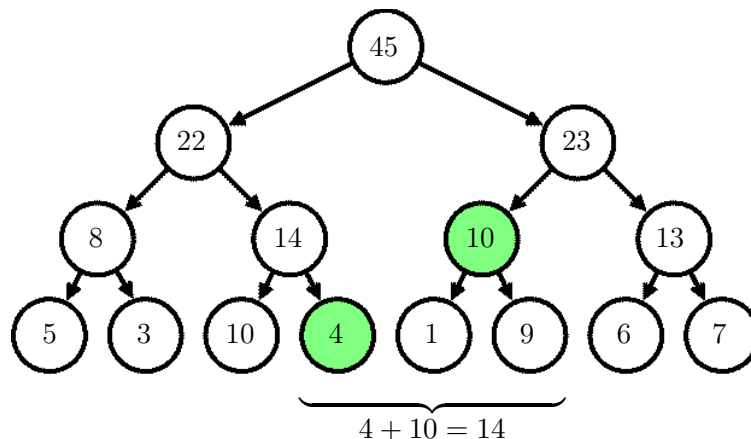
Началното състояние е оцветено с лилаво и е обозначено с  $t_0$ ; финалните състояния са зелени. Автоматът разпознава само наставките на дадения низ, включително целия низ: *a*, *ba* и *aba*. Ако и празния низ смятаме за наставка, то и  $t_0$  трябва да е финално състояние.

Картинката е взета от статията *Suffix Automaton* от сайта *CP-Algorithms*.

**Задача 7.** Сегментното дърво на дадения числов масив  $A = (5; 3; 10; 4; 1; 9; 6; 7)$  изглежда по следния начин:



Сбора  $4 + 1 + 9 = 14$  от елементите на оградения подмасив намираме, събирайки числата от двата зелени върха на сегментното дърво:



**Задача 8.** Скобите на редовете № 76 и № 79 създават блок от оператори (№ 77 и № 78), който привидно е безсмислен, тъй като на това място няма нито цикъл, нито условен оператор, нито оператор за избор на вариант.

```
63         while (true)
64         {
65             int input;
66             cin >> input;
67             if (input >= 0)
68             {
69                 std::lock_guard<std::mutex> guard(workProtect);
70                 tasks.push(input);
71             }
72             else
73                 break;
74             cv.notify_one();
75         }
76         {
77             std::lock_guard<std::mutex> guard(workProtect);
78             isWorking = false;
79         }
80         cv.notify_one();
81         th.join();
82         std::cout << "Done!" << std::endl;
```

Всъщност скобите на редовете № 76 и № 79 определят обхвата на променливата `guard`, декларирана на ред № 77. При изпълнение на ред № 77 се заделя памет за променливата `guard` и се извиква нейният конструктор, който на свой ред извиква метода `lock` на обекта `workProtect`. Това гарантира, че обектът `workProtect` е заключен преди изпълнението на ред № 78 от кода. После програмата напуска блока през ред № 79 и така излиза от обхвата на променливата `guard`. Това води до освобождаване на паметта, заета от променливата, като непосредствено преди това се извиква нейният деструктор, който чрез обръщение към метода `unlock` на обекта `workProtect` премахва заключването.

И така, смисълът на изкуствения блок, създаден чрез скобите на редовете № 76 и № 79, е в своевременното премахване на заключването.

Можем да си спестим променливата `guard` и изкуствения блок от оператори, като в явен вид извикваме методите `lock` и `unlock` на обекта `workProtect` (на редовете № 77 и № 79 съответно). Но така има опасност да не успеем да премахнем заключването: ако на ред № 78 възникне грешка, то програмата няма да стигне до извикването на метода `unlock`. (Ред № 78 е твърде прост, за да предизвика грешка, но там би могло да има някаква по-сложна обработка на данни.)

Оригиналният вариант (този с променливата `guard` и с изкуствения блок от оператори) е защитен от грешки: дори да възникне грешка на ред № 78, програмата ще излезе от блока, извиквайки деструктора на обекта `guard`, който ще извика метода `unlock` на обекта `workProtect` и така ще премахне заключването.

Програмният код на Си-плюс-плюс в задача 8 и в задача 1 е взет от Интернет — съответно от файловете [https://github.com/Angeld55/Data\\_structures\\_and\\_algorithms\\_FMI/blob/main/advanced\\_Multithreading/Async\\_fibb.cpp](https://github.com/Angeld55/Data_structures_and_algorithms_FMI/blob/main/advanced_Multithreading/Async_fibb.cpp) и [https://github.com/Angeld55/Data\\_structures\\_and\\_algorithms\\_FMI/blob/main/advanced\\_CombinatorialGeneration/Combinations.cpp](https://github.com/Angeld55/Data_structures_and_algorithms_FMI/blob/main/advanced_CombinatorialGeneration/Combinations.cpp).