

Що реалізує машина Тьюрінга?

Що реалізує така машина Тьюрінга?

	Λ	1
q_0	R	1 q_1 R
q_1	1 q_F	R

Select one:

- ☐ функцію додавання двох чисел в унарній системі числення
- ☐ нуль-функцію
- ☐ функцію вибору аргументу
- ☒ функцію наступності

Що означає кожна з наведених команд машини Поста?

Що означає кожна з наведених команд машини Поста?

- $\rightarrow i$ *back* змістити активну комірку на одну позицію вліво і перейти до виконання i -ї команди ⇅
- $\leftarrow i$ *back* змістити активну комірку на одну позицію вправо і перейти до виконання i -ї команди ⇅
- ! зупинка, закінчення роботи алгоритму ⇅

Скласти нормальний алгоритм, що виконує збільшення унарного числа на 1

Скласти нормальний алгоритм, що виконує збільшення унарного числа на 1.



Що є різновидом машини Тьюрінга?

Що з наступного є різновидом машини Тьюрінга?

Select one or more:

- ☐ багатостанова машина Тьюрінга
- ☒ багатоголовкова машина Тьюрінга
- ☒ багатострічкова машина Тьюрінга
- ☒ Машина Тьюрінга з багатофункціональним механізмом
- ☒ машина Тьюрінга з багатоповерховою стрічкою

Алгоритм сортування бульбашкою (bubble)

Перетягніть фрагменти коду у вільні місця, щоб отримати алгоритм сортування бульбашкою

```
int temp;
bool wasSwapped;
void bubbleSort(int* array, int size)
{
    for (int i = 1; i <= size; ++i)
    {
        wasSwapped = false;
        for (int j = 0; j < (size - i); ++j)
        {
            for (int i = 1; (i <= size) && wasSwapped; ++i)
            {
                if (array[j] > array[j + 1])
                {
                    temp=array[j];
                    array[j]= array[j + 1];
                    array[j+1]=temp;
                    wasSwapped = true;
                }
            }
        }
    }
}
```

3

for (int j = 0; j < (size - i); ++j)	for (int i = 1; i <= size; ++i)	for (int j = 0; j < size; ++j)
wasSwapped = true;	for (int i = 1; (i <= size) && wasSwapped; ++i)	wasSwapped = false;

```
3 def bubble_sort(array):
4     length = len(array)
5     for i in range(0, length):
6         for j in range(0, length - i - 1):
7             if array[j] > array[j + 1]:
8                 temp = array[j]
9                 array[j] = array[j + 1]
10                array[j + 1] = temp
11
```

Алгоритм сортування вибором(choices)

Перетягніть фрагменти коду у вільні місця, щоб отримати алгоритм сортування вибором

Time left 0:22

```
void choicesSort(int* array, int size)
```

```
{
```

```
    int max, temp;
```

```
    for (int i = 0; i < size-1; i++)
```

```
    {
```

```
        for (int j = i + 1; j < size; j++)
```

```
            max = i;
```

```
            {
```

```
                if (array[j] < array[max])
```

```
                {
```

```
                    max = j;
```

```
                }
```

```
            }
```

```
            temp = array[i]; array[i] = array[max]; array[max] = temp;
```

```
    }
```

```
}
```

```
for (int j = i + 1; j < size; j++)
```

```
max = i;
```

```
max=0;
```

```
for (int j=0; j < size; j++)
```

```
temp = array[i]; array[i] = array[max]; array[max] = temp;
```

```
max = j;
```

```
for (int i = 0; i < size-1; i++)
```

```
def choicesSort(arr):  
    for i in range(len(arr)):  
        for j in range(len(arr) - i - 1):  
            if arr[j] > arr[j + 1]:  
                arr[j], arr[j + 1] = arr[j + 1], arr[j]  
    return arr
```

Алгоритм сортування підрахунком(counting)

Перетягніть фрагменти коду у вільні місця, щоб отримати алгоритм сортування підрахунком

```
void CountingSort(int arr[], int sz) {
```

```
    int i, j, k;
```

```
    int idx = 0;
```

```
    int min, max;
```

```
    k = max - min + 1;
```

```
    for (i = 1; i < sz; i++) {
```

```
        B[arr[i] - min]++;
```

```
    }
```

```
    min = max = arr[0];
```

```
    int *B = new int[k];
```

```
    for (i = 0; i < k; i++) {
```

```
        min = max = 0;
```

```
    }
```

```
    int idx = 0;
```

```
    for (i = 0; i < sz; i++) {
```

```
        arr[idx++] = i;
```

```
    }
```

```
    for (i = min; i <= max; i++)
```

```
        for (j = 0; j < B[i - min]; j++) {
```

```
            min = (min > arr[i]) ? arr[i] : min; max = (arr[i] > max) ? arr[i] : max;
```

```
        }
```

```
    delete[] B;
```

```
}
```

```
    k=0;
```

```
    min = (min > arr[i]) ? arr[i] : min; max = (arr[i] > max) ? arr[i] : max;
```

```
    min=max=0;
```

```
    k = max - min + 1;
```

```
    min = max = arr[0];
```

```
    arr[idx++] = i;
```

```
    B[arr[i] - min]++;
```

```
    B[i] = 0;
```

Вікно > Untitled-1.py > countingSort

```
1  def countingSort(inputArray):
2
3      maxElement= max(inputArray)
4
5      countArrayLength = maxElement+1
6
7      countArray = [0] * countArrayLength
8
9      for el in inputArray:
10         countArray[el] += 1
11
12     for i in range(1, countArrayLength):
13         countArray[i] += countArray[i-1]
14
15     outputArray = [0] * len(inputArray)
16     i = len(inputArray) - 1
17     while i >= 0:
18         currentEl = inputArray[i]
19         countArray[currentEl] -= 1
20         newPosition = countArray[currentEl]
21         outputArray[newPosition] = currentEl
22         i -= 1
23
24     return outputArray
```

Що виконає програма для машини Поста?

Що виконає наступна програма для машини Поста, якщо на стрічці задано послідовність відмічених комірок і початково каретка знаходиться над однією з відмічених комірок?

1. ? 2, 3
2. $\in 4$
3. $\rightarrow 1$
4. !

Select one:

- ☒ програма виконає недопустиму команду і зупиниться
- ☐ подвоює кількість відмічених комірок
- ☐ збільшує кількість відмічених комірок на 1
- ☐ програма зациклиться
- ☐ стирає крайню праву відмічену комірку

[Clear my choice](#)

Задана початкова конфігурація і програма для деякої машини Тьюрінга. Вказати вірні твердження стосовно даної машини.

Нехай задана початкова конфігурація і програма для деякої машини Тьюрінга:

...	Λ	Λ	1	1	1	1	1	Λ	Λ	Λ
-----	-----------	-----------	---	---	---	---	---	-----------	-----------	-----------

\uparrow q_0

$1q_0 \rightarrow \Lambda q_1 R$
 $1q_1 \rightarrow \Lambda q_2 R$
 $\Lambda q_1 \rightarrow 0q_F S$
 $1q_2 \rightarrow \Lambda q_1 R$
 $\Lambda q_2 \rightarrow 1q_F S$

Вкажіть які з наступних тверджень є вірні стосовно даної машини.

Select one or more:

- ☐ дана програма замінює символи 1 на 0 і символи 0 на 1
- ☐ дана програма залишить певну послідовність символів 0 або 1 після її застосування до будьякого слова в алфавіті $\{0,1\}$
- ☒ дана програма залишить один символ 0 або 1 після її застосування до будьякого слова в алфавіті $\{0,1\}$
- ☐ дана програма визначає парність числа, записаного в унарній системі числення
- ☐ дана програма зациклиться

Алгоритм сортування вставкою (insertion)

Перетягніть фрагменти коду у вільні місця, щоб отримати алгоритм сортування вставкою

```
void insertionSort(int* arr, int n) {
```

```
    int temp, j;
```

```
    for (int i = 1; i <= n - 1; i++)
```

```
    {
```

```
        temp = arr[i];
```

```
        j = i - 1;
```

```
        for(j; j >= 0; j--)
```

```
        {
```

```
            arr[j + 1] = arr[j];
```

```
            j = j - 1;
```

```
        }
```

```
        arr[j + 1] = temp;
```

```
    }
```

```
}
```

```
while ((arr[j] > temp) && (j >= 0))
```

```
temp = 0;
```

Побудуйте нормальний алгоритм, який вилучає всі входження, крім першого, символу 'a' з заданого рядка в алфавіті $M = \{a, b\}$

Побудуйте нормальний алгоритм, який вилучає всі входження, крім першого, символу 'a' з заданого рядка в алфавіті $M = \{a, b\}$.

a -> a

*b -> *

* -> .

a* -> *

*a * a*

b b *

* . a

a a* *

Система Поста еквівалентна

Система Поста еквівалентна ...

Select one or more:



системі Тюрінга



системі Гауса



системі Черча-Кліні



системі Маркова



жодній системі

Нормальний алгоритм незастосовний до заданого вхідного слова, якщо на даному кроці

Нормальний алгоритм незастосовний до заданого вхідного слова, якщо на даному кроці

Select one:

- ☒ Зрозуміло, що процес підстановок не зможе зупинитися
- ☐ жодна підстановка не підходить
- ☐ Застосована остання підстановка зі списку підстановок, що задають даний алгоритм
- ☐ Застосована заключна підстановка

Чи може заключний стан зустрічатися в правій частині команди машини Тьюрінга?

Чи може заключний стан зустрічатися в правій частині команди машини Тьюрінга?

Select one:

- ☒ так, це можливо
- ☐ ні, це неможливо
- ☐ так, але тільки для однострічкових машин

Яким буде результат застосування нормального алгоритму в алфавіті $A = \{a, b, c\}$

Яким буде результат застосування нормального алгоритму в алфавіті $A = \{a, b, c\}$:

$cb \rightarrow abc$

$bac \rightarrow ac$

$cab \rightarrow b$

до слова $R = bcabacab$

Answer:

Які два види продукції можливі в нормальних алгоритмах Маркова?

Які два види продукції можливі в нормальних алгоритмах Маркова?

Select one or more:

- ☐ початкова продукція
- ☒ проста продукція
- ☒ заключна продукція
- ☐ складна продукція

Побудуйте нормальний алгоритм, який вилучає всі входження, крім останнього, символу 'a' з заданого рядка в алфавіті $M=\{a,b,c\}$

Побудуйте нормальний алгоритм, який вилучає всі входження, крім останнього, символу 'a' з заданого рядка в алфавіті $M=\{a,b,c\}$.

$ab \rightarrow b$
 $ac \rightarrow c$
 $aa \rightarrow a$

$'bacaabaa' \Rightarrow 'bacabaa' \Rightarrow 'bacbaa' \Rightarrow 'bcbaa' \Rightarrow 'bcba'$

$ab \quad b \quad ba$
 $ac \quad c \quad ca$
 $aa \quad a \quad a$

Яким буде результат застосування нормального алгоритму в алфавіті $A=\{a,b,c\}$

Яким буде результат застосування нормального алгоритму в алфавіті $A = \{a, b, c\}$:

$abc \rightarrow c$
 $ba \rightarrow cb$
 $ca \rightarrow ab$

до слова $R = bacaabc$

1. $bacaabc$
 2. $baacac$
 3. $cbacac$
 4. $cbabac$
 5. $cbac$

Чому дорівнює результат розгалуження $F(ba)$?

Нехай алгоритми A , B і C задані такими правилами:

$A: ab \rightarrow bb; ba \rightarrow aa,$
 $B: ab \rightarrow aba; \underline{ba \rightarrow bab}$
 $C: ab \rightarrow a; \underline{ba \rightarrow b},$
 $R = a.$

Чому дорівнюватиме результат розгалуження $F(ba)$?

☐ bb
☐ aba
☒ bab
☐ aa
☐ a

гл. C,
 якщо $R_1 \rightarrow A$
 якщо на $R_1 \rightarrow B$
 не $R_1 \rightarrow C$

З допомогою якого оператора утворена наступна часткова функція? (μ)

З допомогою якого оператора утворена наступна часткова функція?

$$\mu_y(f(x_1, \dots, x_{n-1}, y) = x_n)$$

Select one:

- ☐ Оператора слабкої мінімізації
- ☒ Оператора мінімізації
- ☐ Оператора суперпозиції
- ☐ Оператора примітивної рекурсії

Запишіть результат виконання найпростіших функцій

Запишіть результати виконання
найпростіших функцій:

$$s^1(3) = 3 + 1 = 4$$

$$o^5(1, 2, 3, 4, 5) = 0$$

$$I_2^3(1, 2, 3) = 2$$

У яких з алгоритмів сортування найгірша оцінка часової складності є $O(n^2)$?

У яких з цих алгоритмів сортування
найгірша оцінка часової складності є
 $O(n^2)$?

Select one or more:

- ☐ Сортування злиттям
- ☒ Сортування вибором
- ☒ Швидке сортування
- ☒ Сортування вставкою
- ☐ Випадкове сортування

За час $O(n^2)$

- Сортування вибором — (англ. *Selection sort*) — пошук найменшого або найбільшого елементу в масиві та його обмін з першим елементом.
- Сортування вставкою (включенням) — (англ. *Insertion sort*) — Визначаємо місце для нового елементу в масиві та вставляємо його.
- Сортування обміном (сортування бульбашкою, англ. *Bubble sort*) — для кожного елементу масиву знаходимо найбільший елемент та обмінюємо його з першим елементом.
- Сортування методом бінарної вставки

Що виконає програма для машини Поста?

Що виконає наступна програма для машини Поста, якщо на стрічці задана послідовність відмічених комірок і початково каретка знаходиться над однією з відмічених комірок?

1. ? 2, 3
2. V 4
3. → 1
4. → 5
5. V 6
6. !

Select one:

- ☐ програма виконає недопустиму команду і зупиниться
- ☐ подвоює кількість відмічених комірок
- ☒ збільшує кількість відмічених комірок на 2
- ☐ програма зациклиться
- ☐ стирає дві відмічені комірки зправа

Який з цих алгоритмів сортування вимагають більше ніж $O(1)$ додаткової пам'яті?

Які з цих алгоритмів сортування вимагають більше ніж $O(1)$ додаткової пам'яті?

Select one or more:

- ☐ Сортування вибором
- ☒ Сортування підрахунком
- ☐ Сортування корзинами
- ☒ Сортування злиттям
- ☒ Сортування вставкою

Внутрішнім алфавітом машини Тьюрінга називають

Внутрішнім алфавітом машини Тьюрінга називають

Select one:

- ☐ символи, які можуть бути записані на стрічці
- ☒ множина станів машини
- ☐ множина команд машини
- ☒ символи, які допускає задача

Clear my choice

Виберіть середню оцінку часової складності для кожного з алгоритмів сортування

Виберіть середню оцінку часової складності для кожного з алгоритмів сортування

Сортування вибором	<input type="text" value="O(n^2)"/>
Сортування вставкою	<input type="text" value="O(n^2)"/>
Швидке сортування	<input type="text" value="O(n log n)"/>
Сортування злиттям	<input type="text" value="O(n log n)"/>

Сортування підрахунком є найбільш ефективним для наборів даних

Сортування підрахунком є найбільш ефективним для наборів даних -

будь-яких

цілих чисел

чисел з вузького діапазону

чисел, що рівномірно розподілені по діапазону

З допомогою якого оператора утворена наступна часткова функція? (всюди визначена)

З допомогою якого оператора утворена наступна часткова функція?

$$M^t f = \begin{cases} Mf, & \text{якщо } Mf \text{ всюди визначена;} \\ \text{не визначена,} & \text{якщо } Mf \text{ визначена не всюди.} \end{cases}$$

Select one:

- ☐ Оператора суперпозиції
- ☐ Оператора примітивної рекурсії
- ☒ Оператора слабкої мінімізації
- ☐ Оператора мінімізації

Які з наступних характеристик притаманні найпростішим рекурсивним функціям?

Які з наступних характеристик притаманні найпростішим рекурсивним функціям?

Select one or more:

- ☐ Найпростіші функції визначені на множині цілих чисел
- ☒ Найпростіші функції є всюди визначені
- ☐ Найпростіші функції можуть бути визначені через інші найпростіші функції
- ☒ Є три найпростіші функції
- ☐ Є дві найпростіші функції
- ☐ Найпростіші функції - це числові функції

Запишіть результати виконання операторів над такими функціями

Запишіть результати виконання операторів над такими функціями:
 $h(x)=0$, $f(x)=x+2$.

$S^2(f,h)=$ $= f(h(x))$

$S^2(f, S^2(f,h))=$ $= f(2)$

$S^2(f, S^2(h,h))=$ $= f(0)$

Побудувати нормальний алгоритм Маркова, який реалізує віднімання $A-B$, де значення A і B є натуральними числами, поданими як ланцюжок символів '1'

Побудувати нормальний алгоритм Маркова, який реалізує віднімання $A-B$, де значення A і B є натуральними числами, поданими як ланцюжки символів '1' (у так званій унарній системі).

1-1 \rightarrow -

-λ \rightarrow λ

1-1 - -1 1 1- -λ λ λ-

-λ -λ -1 λ- 1-1 1- 1

Будь-який алгоритм, заданий у довільній формі, можна замінити еквівалентною йому МТ

Виберіть правильні твердження стосовно наступного формулювання:
Будь-який алгоритм, заданий у довільній формі, можна замінити еквівалентною йому Машиною Тьюрінга.

Select one or more:

- ☒ це твердження неможливо довести
- ☒ Це твердження встановлює відповідність між інтуїтивним поняттям алгоритму і точним математичним поняттям функції, обчислюваної на МТ
- ☐ цю теорему довів Тьюрінг
- ☒ це теза Тьюрінга
- ☐ це теза Черча

Чому дорівнює результат розгалуження $F(ab)$?

Нехай алгоритми A , B і C задані такими правилами:

$A: ab \rightarrow bb; ba \rightarrow aa$,
 $B: ab \rightarrow aba; ba \rightarrow bab$,
 $C: ab \rightarrow a; ba \rightarrow b$,
 $R = A$.

Чому дорівнюватиме результат розгалуження $F(ab)$?

- ☐ a
- ☐ aba
- ☐ aa
- ☒ bb
- ☐ bab

Алгоритм швидкого сортування (quick)

Перетягніть фрагменти коду у вільні місця, щоб отримати алгоритм швидкого сортування (quicksort)

```
void quickSort(int arr[], int left, int right) {  
    int i = left, j = right;  
    int tmp;  
    int p = arr[(left+right)/2];  
    while (i <= j) {  
        while (arr[i] < p) {  
            i++;  
        }  
        while (arr[j] > p) {  
            j--;  
        }  
        if (i <= j) {  
            tmp = arr[i];  
            arr[i] = arr[j];  
            arr[j] = tmp;  
            i++;  
        }  
    }  
}
```

Скільки команд є в машини Поста?

Скільки команд є в машини Поста?

Виберіть одну відповідь:

- ☐ 7
- ☒ 6
- ☐ 5
- ☐ 8
- ☐ Кількість команд необмежена

[Clear my choice](#)

Часова складність алгоритму Швидкого сортування у середньому і в найгіршому випадку

Часова складність алгоритму **Швидкого сортування** у середньому - , а в найгіршому випадку .

Виберіть середню оцінку часової складності для кожного з алгоритмів сортування

Виберіть середню оцінку часової складності для кожного з алгоритмів сортування

Випадкове сортування	Вибрати... n^2
Сортування бульбашкою	Вибрати... n^2
Сортування за розрядами	Вибрати... n
Пірамідальне сортування	Вибрати... $n \log n$

Яким буде результат застосування нормального алгоритма до слова $R=bbaabab$

Яким буде результат застосування нормального алгоритму

$ab \rightarrow \lambda$
 $ba \rightarrow ab$
до слова $R = bbaabab$

Відповідь:

Чи можуть дві різні команди машини Тьюрінга мати однакові ліві частини?

Чи можуть дві різні команди машини Тьюрінга мати однакові ліві частини?

Виберіть одну відповідь:

☐ так, але тільки для однострічкових машин

☐ так, це можливо

☒ ні, це неможливо

Що буде результатом застосування нормального алгоритма до слова $R=abbc$

Що буде результатом застосування нормального алгоритма

$ab \rightarrow bd$
 $db \rightarrow ba$
 $bba \rightarrow abb$
 $c \rightarrow \lambda$
до слова $R = abbc$

Виберіть одну відповідь:

☒ Алгоритм не може бути застосовним до цього слова

☐ bb

☐ aa

☐ cc

Вкажіть результат ітерації алгоритмів на слові babbab

Задано алгоритми A, B і фіксоване слово R (P_1, P_2, P - довільні слова з алфавіту $\{a, b\}$):

$A: P_1abP_2 \rightarrow P_2abP_1,$

$B: P_1bbbP_2 \rightarrow P_1b; P \rightarrow P,$

$R = abb.$

Вкажіть результат ітерації алгоритмів на слові babbab.

- ☐ abbbb
- ☐ ababab
- ☐ abbbab

☒ ітерація повторюватиметься нескінченну кількість разів

Скласти нормальний алгоритм, що в рядку символів з алфавіту $\{a, b, c\}$ здійснює таку заміну

Скласти нормальний алгоритм, що в рядку символів з алфавіту $\{a, b, c\}$ здійснює таку заміну: всі символи 'a' міняє на символ 'b', всі символи 'b' міняє на символ 'c', а всі символи 'c' міняє на 'a'. Наприклад: abbcaacbbba \rightarrow bccabbaccb

$\lambda a \rightarrow^* a$
 $\lambda b \rightarrow^* b$
 $\lambda c \rightarrow^* c$
 $*a \rightarrow b^*$
 $* \rightarrow ,$
 $*b \rightarrow c^*$
 $*c \rightarrow a^*$
 $\lambda a \rightarrow^* a$ $*a \rightarrow b^*$ $* \rightarrow ,$ $a \rightarrow b$ $a \rightarrow^* a$
 $\lambda b \rightarrow^* b$ $*b \rightarrow c^*$ $b \rightarrow c$ $b \rightarrow^* b$
 $\lambda c \rightarrow^* c$ $*c \rightarrow a^*$ $c \rightarrow a$ $c \rightarrow^* c$

Функції, які отримують з функцій системи σ і найпростіших функцій із застосуванням скінченної кількості операторів суперпозиції, примітивної рекурсії та слабкої мінімізації, називають _____ відносно системи σ .

Часткові функції, які отримують з функцій системи σ і найпростіших функцій із застосуванням скінченної кількості операторів суперпозиції, примітивної рекурсії та слабкої мінімізації, називають _____ відносно системи σ .

Select one:

- ☒ примітивно рекурсивними
- ☐ елементарними
- ☐ рекурсивними
- ☐ загальнорекурсивними
- ☐ частково рекурсивними

Побудувати нормальний алгоритм Маркова, який реалізовує додавання $A+B$

Побудувати нормальний алгоритм Маркова, який реалізує додавання $A+B$, де значення A і B є натуральними числами, подані ланцюжки символів '1' (у так званій унарній системі).

$+1 \rightarrow 1+$

$1+ \rightarrow \dots$

$+1$ $+$ λ

$1+$ 1 $1+1$

Please put an answer in each box.

Нехай задано початкова конфігурація і програма для деякої машини Тьюрінга. Яким результатом буде застосування цієї машини до зображеного на стрічці слова?

Нехай задана початкова конфігурація і програма для деякої машини Тьюрінга:

...	Λ	Λ	1	1	1	1	1	1	*	1	1	1	1	Λ	Λ	Λ	...
-----	-----------	-----------	---	---	---	---	---	---	---	---	---	---	---	-----------	-----------	-----------	-----

↑ q_0

$1q_0 \rightarrow \Lambda q_1 R$

$1q_1 \rightarrow 1q_1 R$

$*q_1 \rightarrow 1q_1 R$

$\Lambda q_1 \rightarrow *q_1 S$

Яким результатом буде застосування цієї машини до зображеного на стрічці слова?

Select one:

- ☐ 11111111111*
- ☐ 1111111111
- ☐ 111111*1111*
- ☒ 1111111111*

Що таке Універсальний нормальний алгоритм?

Що таке Універсальний нормальний алгоритм?

Select one:



Це алгоритм, здатний виконувати роботу довільного нормального алгоритму.



Це будьякий нормальний алгоритм.



Це алгоритм, здатний виконувати роботу довільного алгоритму.



Це нормальний алгоритм, який можна зобразити граф-схемою.

[Clear my choice](#)