МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Програмна інженерія та інформаційні технології управління»

Розрахункове завдання з дисципліни «Алгоритми і структури даних»

(Варіант №3)

Виконав:

Студент групи КН-36а

Бодня Є.В.

Перевірила:

Стратієнко Н.К

Харків 2017

**Задание №1**

а)

1. Данный алгоритм представляет собой алгоритм сортировки пузырьком.

В данной сортировке выполняются всего два различных вида операции: сравнение элементов и их обмен.

Так как в алгоритме меняться местами могут только соседние элементы, то каждый обмен уменьшает количество инверсий на единицу. Следовательно, количество обменов равно количеству инверсий в исходном массиве вне зависимости от реализации сортировки. Максимальное количество инверсий содержится в массиве, элементы которого отсортированы по убыванию. Несложно посчитать, что количество инверсий в таком массиве n\*n/2. Получаем, что T2 = O(n^2).

В неоптимизированной реализации на каждой итерации внутреннего цикла производятся n сравнений, а так как внутренний цикл запускается также n раз, то за весь алгоритм сортировки производятся n^2 сравнений.

В оптимизированной версии точное количество сравнений зависит от исходного массива. Известно, что худший случай равен n\*n/2, а лучший — n.

Следовательно, T1 = O(n^2).

В итоге получаем T = T1 + T2 = O(n^2) + O(n^2) = O(n^2).

Итог: Сложность алгоритма: O(n^2).

2. Данный алгоритм представляет собой алгоритм нахождения дискриминанта (квадратного уравнения).

Он не зависит от количества элементов.

С этого следует что он работает за константное время O(1).

б) Даны две функции:

Вывод: функция растёт быстрее чем функция .

)

в)

Функции одного и того же порядка.

Вывод: .

Вывод: =о()

**Задание №2**

Даны две последовательности:

**addaadfbcc**

**cdabdfadcc**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **a** | **d** | **d** | **a** | **a** | **d** | **f** | **b** | **c** | **c** |
|  | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **c** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **1** |
| **d** | **0** | **0** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** |
| **a** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **2** | **2** | **2** |
| **b** | **0** | **1** | **1** | **1** | **2** | **2** | **2** | **2** | **3** | **3** | **3** |
| **d** | **0** | **1** | **2** | **2** | **2** | **2** | **3** | **3** | **3** | **3** | **3** |
| **f** | **0** | **1** | **2** | **2** | **2** | **2** | **3** | **4** | **4** | **4** | **4** |
| **a** | **0** | **1** | **2** | **2** | **3** | **3** | **3** | **4** | **4** | **4** | **4** |
| **d** | **0** | **1** | **2** | **3** | **3** | **3** | **4** | **4** | **4** | **4** | **4** |
| **c** | **0** | **1** | **2** | **3** | **3** | **3** | **4** | **4** | **4** | **5** | **5** |
| **c** | **0** | **1** | **2** | **3** | **3** | **3** | **4** | **4** | **4** | **5** | **6** |

*Наибольшая общая подпоследовательность:* **dadfcc**

Даны две последовательности:

**dfaedfcfba**

**abbaedbfba**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **d** | **f** | **a** | **e** | **d** | **f** | **c** | **f** | **b** | **a** |
|  | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **a** | **0** | **0** | **0** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** |
| **b** | **0** | **0** | **0** | **1** | **1** | **1** | **1** | **1** | **1** | **2** | **2** |
| **b** | **0** | **0** | **0** | **1** | **1** | **1** | **1** | **1** | **1** | **2** | **2** |
| **a** | **0** | **0** | **0** | **1** | **1** | **1** | **1** | **1** | **1** | **2** | **3** |
| **e** | **0** | **0** | **0** | **1** | **2** | **2** | **2** | **2** | **2** | **2** | **3** |
| **d** | **0** | **1** | **1** | **1** | **2** | **3** | **3** | **3** | **3** | **3** | **3** |
| **b** | **0** | **1** | **1** | **1** | **2** | **3** | **3** | **3** | **3** | **4** | **4** |
| **f** | **0** | **1** | **2** | **2** | **2** | **3** | **4** | **4** | **4** | **4** | **4** |
| **b** | **0** | **1** | **2** | **2** | **2** | **3** | **4** | **4** | **4** | **5** | **5** |
| **a** | **0** | **1** | **2** | **3** | **3** | **3** | **4** | **4** | **4** | **5** | **6** |

*Наибольшая общая подпоследовательность:* **aedfba**

**Задание №3**

Даны заявки:

[12,15), [1,3), [14,16), [4,5), [10,13), [7,13), [8,10), [5,6), [7,10), [5,8).

[1,3) [5,8) [8,10) [14,16)

[4,5) [7,10) [10,13)

[5,6) [7,13) [12,15)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Даны заявки:

[1,3), [4,6), [7,12), [13,15), [13,16), [8,9), [11,15), [7,11), [11,12), [3,5).

[1,3) [7,11) [11,12) [13,16)

[3,5) [7,12) [11,15)

[4,6) [8,9) [13,15)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

**Задание №4**

Числа для сложения 3+10 = 13 -> 1+3 -> 1 + 111  
Машина Тьюринга реализует алгоритм сложения 2-ух чисел так называемой “наивной” унарной записи. Даны два набора “1”, которые в памяти машины Тьюринга будут разделены “+”.  
Таким образом на ленту подается 1 + 3,  а именно “1”+”111”:  
Работу машины Тьюринга нельзя свести у простому стираю знака + так как возникает пустая ячейка возле “1”, следовательно совокупность ячеек нельзя будет рассматривать как изображение натурального числа.   
Головка машины над крайней слева “1”, обозреваемая “1” стирается происходит сдвиг вправо и переход в состояние q(2).  
Все последующие такты сводятся к сдвигам направо сквозь все “1” и “+” до тех пор пока не будет достигнута первая пустая ячейка.   
В пустую ячейку вписывается “1” и машина переходит в состояние q(1). В результате одна “1” левого слагаемого оказалась перенесенной в правое слагаемое.  
Далее при состоянии q(1) происходит сдвиг влево сквозь все “1” и “+” пока не будет достигнута пустая ячейка.  
Тогда происходит сдвиг вправо и машина переходит в состояние q(0), а в поле зрения устанавливается оставшаяся в левом слагаемом “1”.   
Обозреваемая “1” стирается происходит сдвиг вправо и переход в состояние q(2).   
Все последующие такты сводятся к сдвигам направо сквозь все “1” и “+” до тех пор пока не будет достигнута первая пустая ячейка.   
В пустую ячейку вписывается “1” и машина переходит в состояние q(1). В результате все “1” левого слагаемого оказалась перенесенными в правое слагаемое.  
Далее при состоянии q(1) происходит сдвиг влево сквозь все “1” и “+” пока не будет достигнута пустая ячейка.  
После машина переходит в состояние q(0) с последующим сдвигом вправо. Когда при состоянии q(0) в поле зрения машины попадает знак “+” он удаляется и машина прекращает свою работу.

0 1 2 3 4 5 6 7

…

\_

\_

1

1

1

+

1

\_

…

0 1 2 3 4 5 6 7

…

\_

\_

1

1

1

+

1

\_

…

q0

0 1 2 3 4 5 6 7

…

\_

\_

1

1

1

+

\_

\_

…

q2

…

\_

0 1 2 3 4 5 6 7

\_

1

1

1

+

\_

\_

…

q2

0 1 2 3 4 5 6 7

…

1

\_

1

1

1

+

\_

\_

…

q1

0 1 2 3 4 5 6 7

1

…

\_

1

1

1

+

\_

\_

…

q0

0 1 2 3 4 5 6 7

Результат:

1

…

\_

1

1

1

\_

\_

\_

…

**Задание №5**

1.

***Сортировка пузырьком:***

Дана последовательность:

2 9 1 9 7 2 0 5 7 8

Происходит сравнение элементов каждого с каждым. Если один элемент меньше второго, то меняем их местами.

Начнём с элемента с индексом 1: меняем его местами с элементом с индексом 2.

2 1 9 9 7 2 0 5 7 8

Возьмем элемент с индексом 3 и поменяем его местами с элементом с индексом 4.

2 1 9 7 9 2 0 5 7 8

Повторяем проверку всех следующих элементов.

Итог: за время O() можно получить отсортированную последовательность.

***Сортировка слиянием:***

Дана последовательность:

2 9 1 9 7 2 0 5 7 8

Разобьём эту последовательность на массивы размером 1.

Сольём эти массивы в один массив так, чтобы при сливании элементы сравнивались между собой и занимали правильные места в результирующем массиве.

9

9

9

8

7

7

5

2

2

1

0

7

8

0

1

2

2

5

7

9

7

8

7

5

2

0

9

9

1

2

8

7

5

0

7

2

1

9

2

9

8

7

5

0

2

7

9

1

9

2

***Сортировка кучей:***

Дана последовательность:

2 9 1 9 7 2 0 5 7 8

Построим кучу из этой последовательности:

Начиная с элемента с индексом (n/2 – 1) и до элемента с индексом 0 начнём выбирать максимального наследника и менять его местами с родителем, если родитель меньше максимального наследника.

Таким образом мы получили кучу с максимальным элементом на вершине. Теперь меняем максимальный элемент с последним местами, записывая максимальный элемент в результирующий массив. Эту операцию повторяем до тех пор, пока массив не будет отсортирован.

***Быстрая сортировка:***

Дана последовательность:

2 9 1 9 7 2 0 5 7 8

1. Выберем элемент из массива. Назовём его опорным. Пускай этим опорным элементом будет элемент с индексом (n/2 - 1).

2.Разбиение: будем перераспределять элементы в массиве таким образом, что элементы меньше опорного помещаются перед ним, а больше или равные после.

3.Будем рекурсивно применить первые два шага к двум подмассивам слева и справа от опорного элемента. Рекурсия не применяется к массиву, в котором только один или отсутствуют элементы.

Опорный элемент

7

8

5

0

2

7

9

1

2

9

7

8

5

0

2

7

9

1

2

9

2

7

8

9

0

2

7

9

1

5

2

7

8

9

9

2

7

0

1

5

2

7

8

9

9

7

2

0

1

5

Так как указатели на начало и конец столкнулись, то теперь аналогично нужно отсортировать левую и правую часть массива (относительно среднего элемента массива с индексом n/2).

2.

***Сортировка пузырьком:***

Дана последовательность:

5 9 1 7 1 7 4 8 2 4

Происходит сравнение элементов каждого с каждым. Если один элемент меньше второго, то меняем их местами.

Начнём с элемента с индексом 1: меняем его местами с элементом с индексом 2.

5 1 9 7 1 7 4 8 2 4

Возьмем элемент с индексом 2 и поменяем его местами с элементом с индексом 3.

5 1 7 9 1 7 4 8 2 4

Повторяем проверку всех следующих элементов.

Итог: за время O() можно получить отсортированную последовательность.

***Сортировка слиянием:***

Дана последовательность:

5 9 1 7 1 7 4 8 2 4

Разобьём эту последовательность на массивы размером 1.

Сольём эти массивы в один массив так, чтобы при сливании элементы сравнивались между собой и занимали правильные места в результирующем массиве.

9

9

8

7

7

5

4

4

2

1

1

2

4

1

1

4

5

7

7

8

2

4

8

7

4

1

9

7

1

5

4

2

8

4

7

1

1

7

5

9

4

2

8

4

7

1

7

1

9

5

***Сортировка кучей:***

Дана последовательность:

5 9 1 7 1 7 4 8 2 4

Построим кучу из этой последовательности:

Начиная с элемента с индексом (n/2 – 1) и до элемента с индексом 0 начнём выбирать максимального наследника и менять его местами с родителем, если родитель меньше максимального наследника.

Таким образом мы получили кучу с максимальным элементом на вершине. Теперь меняем максимальный элемент с последним местами, записывая максимальный элемент в результирующий массив. Эту операцию повторяем до тех пор, пока массив не будет отсортирован.

***Быстрая сортировка:***

Дана последовательность:

5 9 1 7 1 7 4 8 2 4

1. Выберем элемент из массива. Назовём его опорным. Пускай этим опорным элементом будет элемент с индексом (n/2 + 1).

2.Разбиение: будем перераспределять элементы в массиве таким образом, что элементы меньше опорного помещаются перед ним, а больше или равные после.

3.Будем рекурсивно применить первые два шага к двум подмассивам слева и справа от опорного элемента. Рекурсия не применяется к массиву, в котором только один или отсутствуют элементы.

Опорный элемент

2

4

8

4

7

1

7

1

5

9

2

4

8

4

7

1

7

1

5

9

2

9

8

4

7

1

7

1

5

4

2

9

8

4

7

1

7

1

5

4

7

9

8

4

7

1

2

1

5

4

7

9

8

4

7

1

2

1

5

4

7

9

8

7

4

1

2

1

5

4

Так как указатели на начало и конец столкнулись, то теперь аналогично нужно отсортировать левую и правую часть массива (относительно среднего элемента массива с индексом n/2).

**Задание №6**

Последовательность, сгенерированная линейным конгруэнтным методом, с коэффициентами a=69069, c=5, m=:

14 15 12 17 14 11 12 13 6 19 8 9 10 19 8 1 2 11 12 5

Переведем последовательность в двоичную систему счисления:

11101111 11001000 11110101 11100110 11101001 11000100 11010100 11100011 01011110 0101

**Частотный побитовый тест:**

n=76;

=14;

=;

P=erfc(;

Последовательность случайная.

**Тест на последовательность одинаковых битов:**

=0.592105;

=38+1=39;

P=erfc( erfc(2.289/0.683)=3.351;

Последовательность случайная.

**Тест на самую длинную последовательность из единиц в блоке:**

=1,7277+1,407+0,0263+0,333=1,7277+1.7663=3,494;

p=0.284958;

Последовательность случайная.

**Вывод:** последовательность полностью случайная.

Последовательность, сгенерированная методом Фибоначчи с запаздываниями, с лагами a=2, b=54:

1 4 26 4 31 96 37 72 20 13 14 19 16 21 14 19 16 21 26 19

Переведем последовательность в двоичную систему счисления:

11001101 01001111 11100000 10010110 01000101 00110111 10100111 00001010 11110100 11100001 01011101 010011

**Частотный побитовый тест:**

n=94;

= 4;

=;

P=erfc(

Последовательность случайная.

**Тест на последовательность одинаковых битов:**

=49/94=0.52;

=47+1=48;

P=erfc(1.07/0.7)=0.0306393;

Последовательность случайная.

**Тест на самую длинную последовательность из единиц в блоке:**

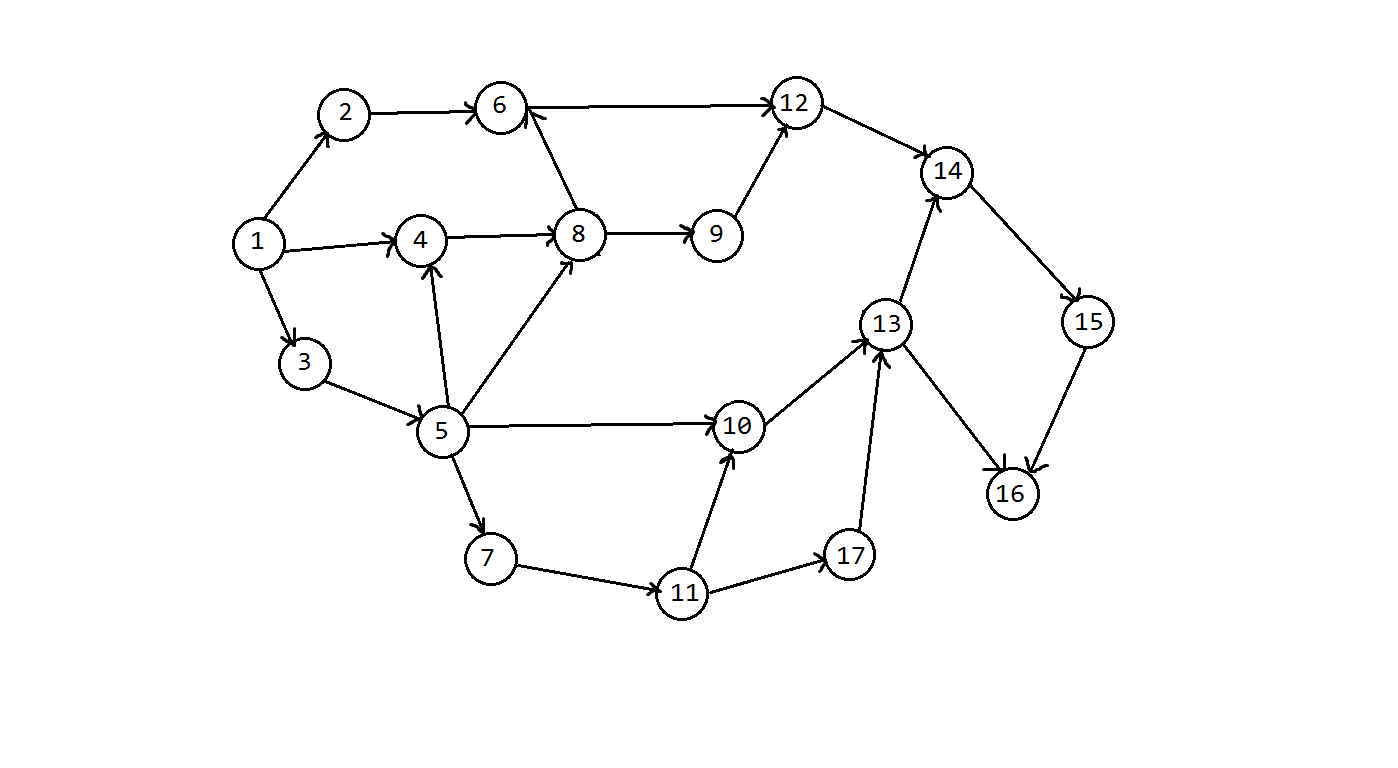
=0,6+1,407+0,466+0,333=1,7277+1.7663=2,806;

p=0.374443;

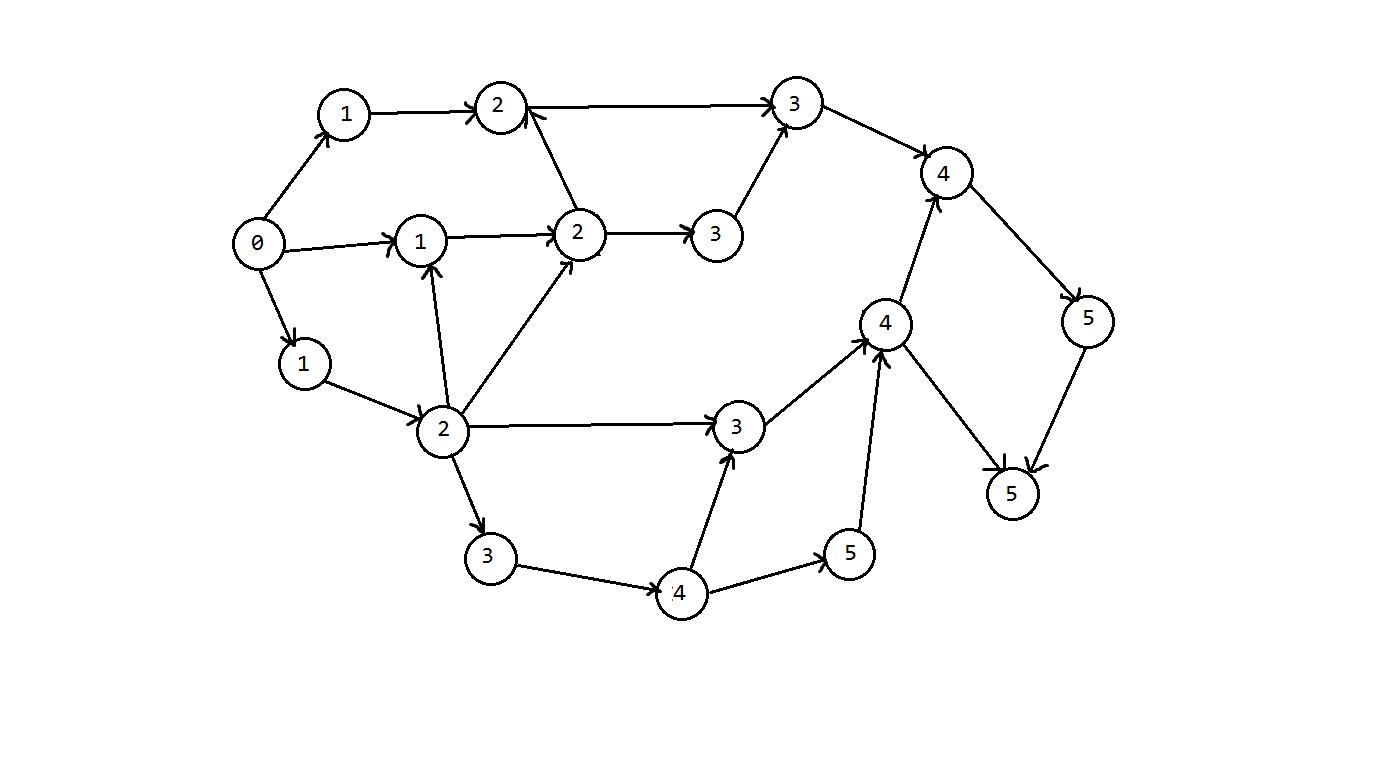
Последовательность случайная.

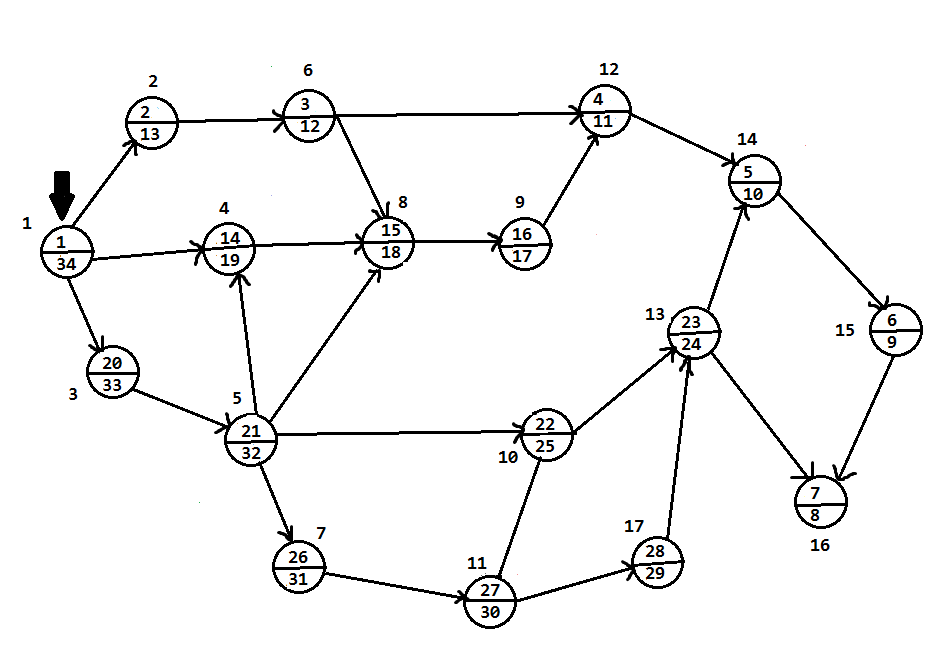
**Вывод:** последовательность полностью случайная.

**Задание №7**

Заданный граф

Поиск в ширину



Поиск в глубину

Топологическая сортировка

Возьмем вершины в порядке выхода:

16 15 14 12 6 2 9 8 4 13 10 17 11 7 5 3 1

Заменив порядок вершин на обратный, мы получим топологическую сортировку:

1 3 5 7 11 17 10 13 4 8 9 2 6 12 14 15 16

**Задание №8**

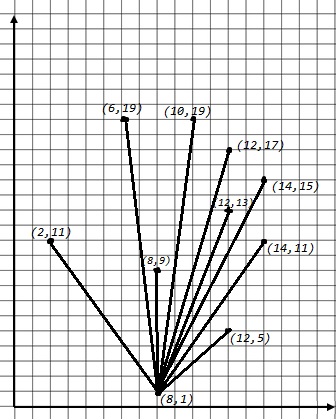
**Дано множество точек:**

(14;15) (12;17) (14;11) (12;13) (6;19) (8;9) (10;19) (8;1) (2;11) (12;5)

Нахождение минимальной точки объекта:

Сортировка всех точек по полярному углу относительно . (8;1)

Сортировка по возрастанию:

(8;1) (12;5) (14;11) (14;15) (12;13) (12;17) (10;19) (8;9) (6;19) (2;11)

Шаг 0. Q={}

Шаг 1. .

() (=(4;4)(6;10)

Q={}

Шаг 2. .

() (=(2;6)(2;10)

Q={}

Шаг 3. .

() (=(0;4)(-2;2)

Q={}

Шаг 4. .

() (=(-2;-2)(-2;2)

Q={}

Шаг 5. .

() (=(-2;2)(-4;4)

Q={}

Шаг 6. .

() (=(-2;2)(-4;-8)

> 0

Q={}

Шаг 7. .

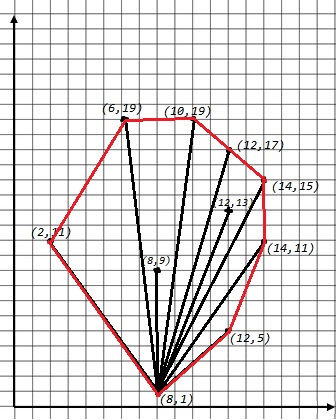
() (=(-2;-10)(-4;0)

Q={}

Шаг 8. .

() (=(-2;10)(-6;2)

Q={}

**Вывод:** Q={}