Жамбыл облысы әкімдігі білім басқармасы

«Білім» кәсіби гуманитарлық-техникалық колледжі

(білім беру ұйымының атауы)

**Оқу сабағының жоспары**

(теориялық немесе өндірістік оқыту)

**Сұрыптау алгоритмдерімен жұмыс.**

(сабақ тақырыбы)

**Модуль/Пән атауы** Бағдарламалық кодты қайта өңдеу

**Дайындаған педагог** Нургисаева У.М

**20\_**25**\_ жылғы** «\_\_\_\_» \_\_\_наурыз\_\_\_

**1. Жалпы мәліметтер**

Курс, оқу жылы, топ 3 курс, 3БҚ-1-22, 3БҚ-2-22

Сабақ түрі: Кіріктірілген

**2. Мақсаты, міндеттері:**

**Оқу:** Білім алушыларға python (пайтон) программалау тілінде сұрыптау алгоритмдерімен жұмыс, есеп шығаруды жүзеге асыру, қолдану жолдарын ұйымдастыру, программаны өңдеу жұмыснегіздерін үйрету

#### **Дамыту:** Логикалық ойлау қабілеттерін дамыту, есептердің әртүрлі шешімдерін табуға дағдыландыру, программалық кодтың құрылымын дұрыс құру қабілеттерін қалыптастыру.

#### **Тәрбиелік:** Жауапкершілік пен ұқыптылыққа тәрбиелеу, алгоритмдік ойлауды дамыту, шығармашылық қабілеттерін жетілдіру.

**3. Оқу-жаттығу процесінде білім алушылар меңгеретін күтілетін нәтижелер және кәсіби дағдылар тізбесі:** Python программалау сұрыптау алгоритмдерімен жұмыс, есеп шығаруды жүзеге асыру, қолдану жолдарын ұйымдастыру, программаны өңдеу және пайдалану дағдыларын меңгеру. Таңдау, көпіршікті, жылдам сұрыптау жолдарының жұмыс негіздерін үйренеді, код жазады.

**4. Қажетті ресурстар:** ДК немесе ноутбуктер

***5. Сабақтың барысы: (90 минут)***

**5.1. Ұйымдастыру кезеңі:** *( 3 мин )*

**5.2. Үй жұмысын жан-жақты тексеру:**

***“Сұрақ-жауап” әдісі*** *(15 минут)*

**Сұрыптау алгоритмдерімен жұмыс.**

Сұрыптау дегеніміз – жиын объектілерін белгілі бір тәртіппен ауыстырып қою процесі. Сұрыптаудың мақсаты - элементтерді сұрыпталған жиында іздеуді жеңілдету. Массивтерді сұрыптау әдістеріне қойылатын негізгі талап – жадыны тиімді пайдалану.

Сұрыптау алгоритмдері көптеген салаларда қолданылады, атап айтқанда:

* Дерекқор жүйелерінде деректерді тиімді іздеу.
* Графикалық интерфейстерде элементтерді реттеу.
* Есептеу математикасында үлкен деректер жиынтықтарын өңдеу.
* Машиналық оқытуда деректерді алдын ала өңдеу.

Python тілінде сұрыптаудың бірнеше әдістері бар. Біз негізгі алгоритмдерді қарастырамыз:

1. Таңдау (Selection Sort)
2. Кірістіру (Insertion Sort)
3. Көпіршікті (Bubble Sort)
4. Жылдам (Quick Sort)
5. Біріктіру (Merge Sort)
6. Қыстырма (Heap Sort)
7. Сандық (Counting Sort)

**3. Негізгі сұрыптау алгоритмдері**

**3.1. Таңдау арқылы сұрыптау (Selection Sort)**

Таңдау сұрыптау – ең қарапайым алгоритмдердің бірі. Ол тізімнің ең кіші элементін тауып, оны бірінші орынға қояды, содан кейін келесі ең кіші элементті тауып, оны екінші орынға қояды және т.б.

def selection\_sort(arr):

for i in range(len(arr)):

min\_idx = i

for j in range(i+1, len(arr)):

if arr[j] < arr[min\_idx]:

min\_idx = j

arr[i], arr[min\_idx] = arr[min\_idx], arr[i]

arr = [64, 25, 12, 22, 11]

selection\_sort(arr)

print("Сұрыпталған массив:", arr)

**3.2. Кірістіру арқылы сұрыптау (Insertion Sort)**

Кірістіру сұрыптау массивті шартты түрде сұрыпталған және сұрыпталмаған екі бөлікке бөледі. Әр жаңа элемент өз орнына кірістіріледі.

def insertion\_sort(arr):

for i in range(1, len(arr)):

key = arr[i]

j = i - 1

while j >= 0 and key < arr[j]:

arr[j + 1] = arr[j]

j -= 1

arr[j + 1] = key

arr = [12, 11, 13, 5, 6]

insertion\_sort(arr)

print("Сұрыпталған массив:", arr)

**3.3. Көпіршікті сұрыптау (Bubble Sort)**

Бұл алгоритм көрші элементтерді салыстырып, оларды орындарын ауыстыру арқылы сұрыптайды.

def bubble\_sort(arr):

n = len(arr)

for i in range(n):

for j in range(0, n-i-1):

if arr[j] > arr[j+1]:

arr[j], arr[j+1] = arr[j+1], arr[j]

arr = [64, 34, 25, 12, 22, 11, 90]

bubble\_sort(arr)

print("Сұрыпталған массив:", arr)

**3.4. Жылдам сұрыптау (Quick Sort)**

Жылдам сұрыптау рекурсиялық алгоритм болып табылады. Ол массивтен **бөлу нүктесін (pivot)** таңдап, сол элементтен кіші және үлкен бөліктерге бөледі.

def quick\_sort(arr):

if len(arr) <= 1:

return arr

pivot = arr[len(arr) // 2]

left = [x for x in arr if x < pivot]

middle = [x for x in arr if x == pivot]

right = [x for x in arr if x > pivot]

return quick\_sort(left) + middle + quick\_sort(right)

arr = [10, 7, 8, 9, 1, 5]

print("Сұрыпталған массив:", quick\_sort(arr))

**3.5. Біріктіру арқылы сұрыптау (Merge Sort)**

Біріктіру сұрыптауы **"Бөліп ал, сұрыпта, біріктір"** стратегиясын қолданады.

def merge\_sort(arr):

if len(arr) > 1:

mid = len(arr) // 2

left = arr[:mid]

right = arr[mid:]

merge\_sort(left)

merge\_sort(right)

i = j = k = 0

while i < len(left) and j < len(right):

if left[i] < right[j]:

arr[k] = left[i]

i += 1

else:

arr[k] = right[j]

j += 1

k += 1

while i < len(left):

arr[k] = left[i]

i += 1

k += 1

while j < len(right):

arr[k] = right[j]

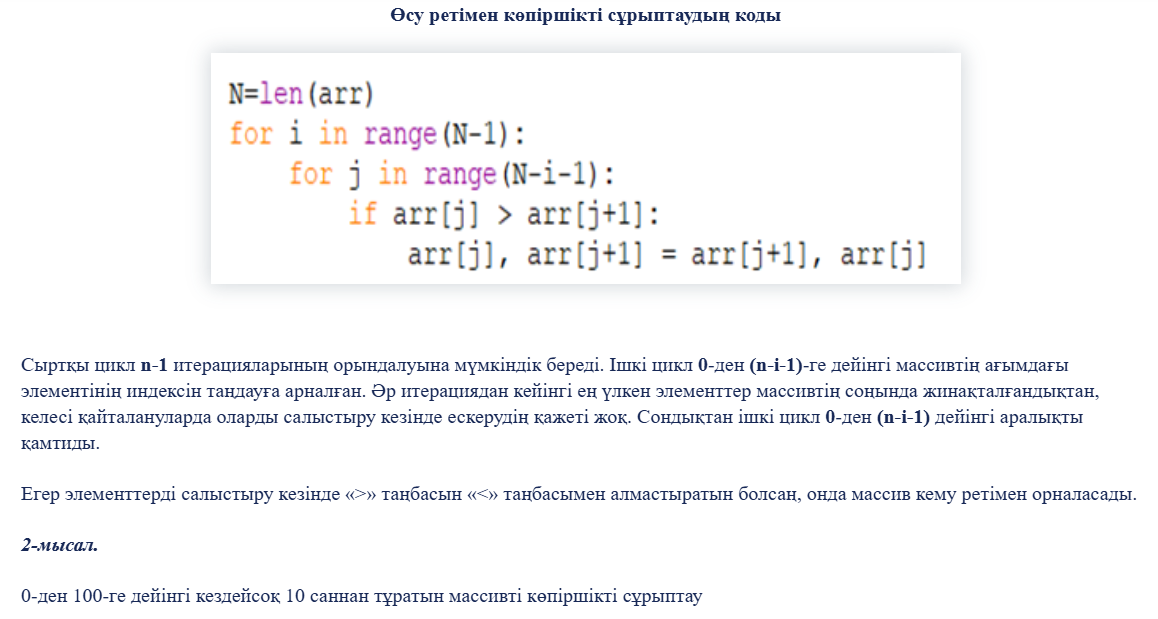
j += 1

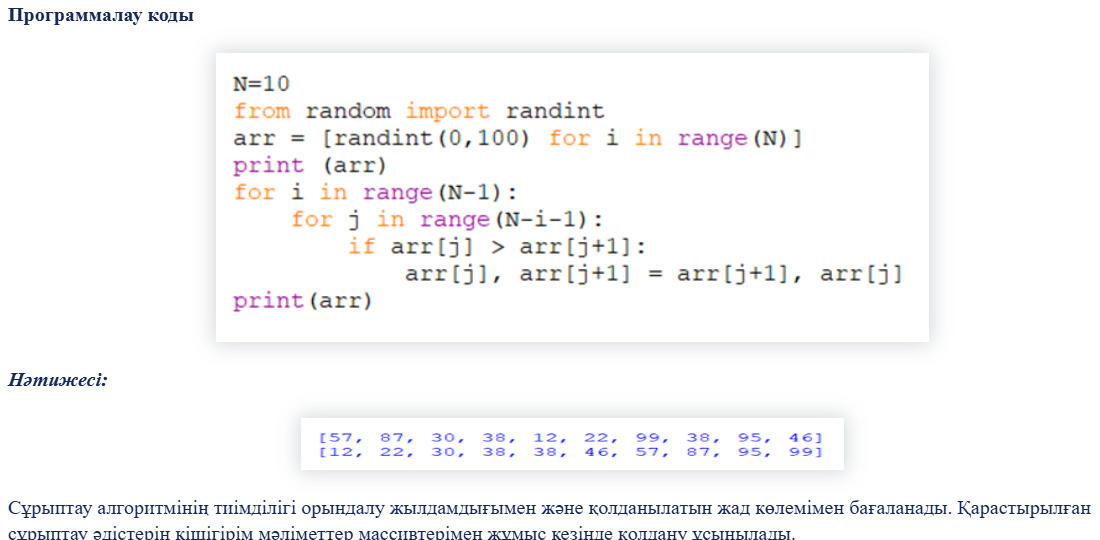
k += 1

arr = [38, 27, 43, 3, 9, 82, 10]

merge\_sort(arr)

print("Сұрыпталған массив:", arr)





**1. Есеп:** **N** бүтін саннан тұратын тізім берілген. Осы тізімді **өсу ретімен** сұрыптап, нәтижені экранға шығарыңыз.

def selection\_sort(arr):

n = len(arr)

for i in range(n):

min\_idx = i

for j in range(i + 1, n):

if arr[j] < arr[min\_idx]:

min\_idx = j

arr[i], arr[min\_idx] = arr[min\_idx], arr[i] # Элементтерді орындарын ауыстыру

return arr

n = int(input("Элементтер саны: "))

arr = list(map(int, input("Сандарды енгізіңіз: ").split()))

sorted\_arr = selection\_sort(arr)

print("Сұрыпталған массив:", \*sorted\_arr)

**2. Есеп:** **N** элементтен тұратын тізім берілген. Осы тізімді **кеміту ретімен** көпіршікті сұрыптау алгоритмін қолдана отырып сұрыптаңыз. Сонымен қатар, әр итерациядан кейінгі аралық нәтижені шығарыңыз.

def bubble\_sort(arr):

n = len(arr)

for i in range(n):

swapped = False

for j in range(0, n - i - 1):

if arr[j] < arr[j + 1]: # Кему ретімен сұрыптау

arr[j], arr[j + 1] = arr[j + 1], arr[j]

swapped = True

print("Кезең:", \*arr) # Әр итерациядан кейін тізімді шығару

if not swapped:

break

n = int(input("Элементтер саны: "))

arr = list(map(int, input("Сандарды енгізіңіз: ").split()))

bubble\_sort(arr)

print("Сұрыпталған массив:", \*arr)

**3. Есеп:** **N** бүтін саннан тұратын тізім берілген. Жылдам сұрыптау (Quick Sort) алгоритмін пайдаланып, осы тізімді **өсу ретімен** сұрыптаңыз.

def quick\_sort(arr):

if len(arr) <= 1:

return arr # Егер бір ғана элемент болса, сұрыптаудың қажеті жоқ

pivot = arr[len(arr) // 2] # Бөлу элементі (ортадағы элемент)

left = [x for x in arr if x < pivot]

middle = [x for x in arr if x == pivot]

right = [x for x in arr if x > pivot]

return quick\_sort(left) + middle + quick\_sort(right)

n = int(input("Элементтер саны: "))

arr = list(map(int, input("Сандарды енгізіңіз: ").split()))

sorted\_arr = quick\_sort(arr)

print("Сұрыпталған массив:", \*sorted\_arr)

|  |  |  |
| --- | --- | --- |
| **Білдім** | **Білемін** | **Білгім келеді** |
|  |  |  |

**Рефлексия**

**Үй тапсырмасы –** Зертханалық жұмыс №8 орындау және қорғау