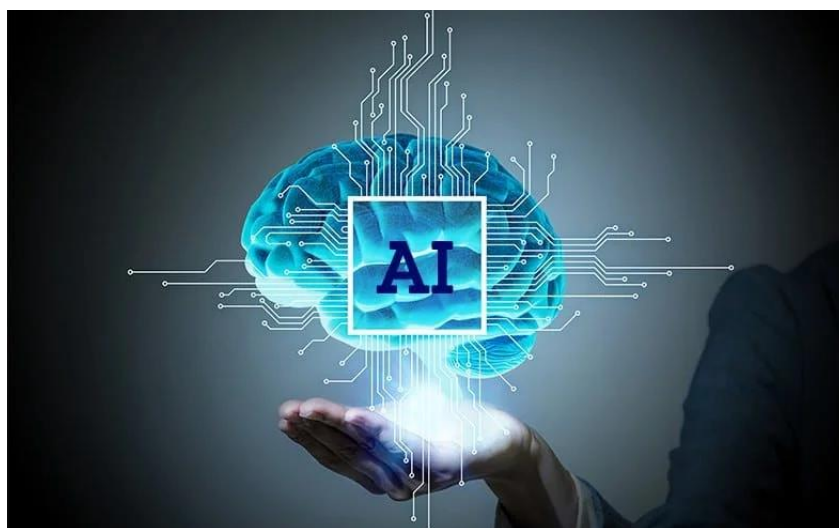


**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
KHOA TOÁN - TIN HỌC**

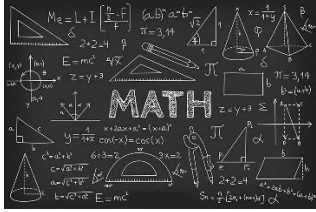


BÀI BÁO CÁO THỰC HÀNH TUẦN 4



MÔN HỌC: Phân Tích Thuật Toán
Sinh Viên: Trần Công Hiếu - 21110294
Lớp: 21TTH

TP.HCM, ngày 28 tháng 04 năm 2024



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM
KHOA TOÁN – TIN HỌC



BÀI BÁO CÁO THỰC HÀNH TUẦN 4

HK1 - NĂM HỌC: 2024-2025

MÔN: PHÂN TÍCH THUẬT TOÁN

SINH VIÊN: TRẦN CÔNG HIẾU

MSSV: 21110294

LỚP: 21TTH

[illegible]

Giảng viên Bộ môn

Mục Lục

Bài 1.	5
1.1. Chương trình với N cố định.	5
1.2. Chương trình với N tăng dần theo thứ tự 10, 20,	7
Bài 2.	9
2.1. Chương trình với N tăng dần dùng hàm có sẵn.	9
2.2. Chương trình với N tăng dần dùng hàm tự định nghĩa.	11
2.3. Chương trình với N cố định và tính thời gian trung bình.	14

Bài 1.

1.1. Chương trình với N cố định.

```
1 import numpy as np
2 def search_x_in_A(A, x):
3     save = []
4     for index, value in enumerate(A):
5         if(value == x):
6             save.append(index)
7     if save: return save
8     else: return "Khong co gia tri nao"
9
```

(File: bai1_1.py).

“import numpy as np”: Gọi thư viện numpy để thao tác với mảng trong đoạn mã của chương trình và gọi tắt là “np”, điều này cho phép gọi các hàm và đối tượng từ thư viện numpy bằng cách sử dụng tiền tố “np”.

“def search_x_in_A(A, x):”: Định nghĩa hàm tìm kiếm x trong mảng A. Với tham số truyền vào là mảng A và giá trị cần tìm kiếm x.

“save = []”: Khởi tạo một list save để lưu trữ các index mà giá trị của mảng A tại các index đó bằng x. Vì đề bài chỉ tạo mảng A nhưng không nói đến từ khóa “khác nhau” điều đó cho thấy có thể tồn tại nhiều vị trí mà tại đó thỏa $A[\text{index}] = x$. Do đó ta khởi tạo list để lưu trữ chúng, thuận tiện cho việc trả ra thông báo.

“for index, value in enumerate()”: Tạo vòng lặp for duyệt từng cặp index, value. Cặp giá trị index, value tương ứng được trả về từ mảng A thông qua phương thức enumerate().

“if(value == x):”: So sánh điều kiện nếu giá trị value bằng giá trị x cần tìm thì thêm index tương ứng với value ấy vào list save thông qua phương thức append().

Cuối cùng kiểm tra “if save: return save” nếu save không rỗng, tức có index mà giá trị của mảng A tại đó bằng x cần tìm kiếm thì ta in list các index thỏa $A[\text{index}] = x$. Ngược lại, nếu save rỗng thì trả về thông báo không có giá trị nào.

```
10 x=10
11 N = 1000
12
13 A = sorted(np.random.randint(0,1001,N))
14 print("- Mang A co",N,"phan tu.")
15 print(A)
16 print(search_x_in_A(A,x))
17
```

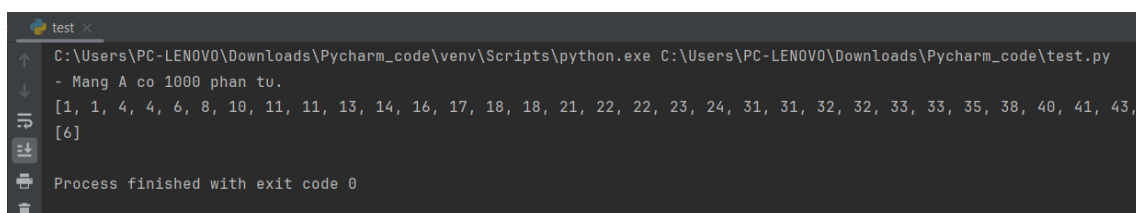
Khởi gán 2 giá trị để chạy thử cho đoạn mã, lần lượt là giá trị cần tìm kiếm x trong mảng A là x = 10 và N = 1000.

“A = sorted(np.random.randint(0,100,N))”: Tạo mảng ngẫu nhiên nguyên với giá trị mỗi phần tử nằm trong khoảng từ 0 đến 1000 với N phần tử. Và sắp xếp mảng thông qua hàm sorted(). Lúc này, A là mảng nguyên được tạo ngẫu nhiên có giá trị từ 0 đến 1000 và được sắp xếp sẵn.

Cuối cùng là print ra thông báo mảng A có N phần tử, sau đó in các giá trị của A để quan sát so với kết quả hàm search_x_in_A(A, x) trả về.

Kết quả.

- Khi tồn tại phần tử trong mảng thỏa $A[i] = x$.



```
test x
C:\Users\PC-LENOVO\Downloads\Pycharm_code\venv\Scripts\python.exe C:\Users\PC-LENOVO\Downloads\Pycharm_code\test.py
- Mang A co 1000 phan tu.
[1, 1, 4, 4, 6, 8, 10, 11, 11, 13, 14, 16, 17, 18, 18, 21, 22, 22, 23, 24, 31, 31, 32, 32, 33, 33, 35, 38, 40, 41, 43,
[6]
Process finished with exit code 0
```

- Khi không tồn tại phần tử trong mảng thỏa $A[i] = x$.

```
test
C:\Users\PC-LENOVO\Downloads\Pycharm_code\venv\Scripts\python.exe C:\Users\PC-LENOVO\Downloads\Pycharm_code\test.py
- Mang A co 1000 phan tu.
[0, 1, 3, 5, 7, 7, 7, 8, 8, 9, 12, 12, 13, 15, 18, 18, 19, 22, 22, 24, 26, 28, 28, 28, 28, 28, 29, 29, 29, 29, 30,
Khong co gia tri nao
Process finished with exit code 0
```

1.2. Chương trình với N tăng dần theo thứ tự 10, 20, ...

Tương tự như trên, hàm `search_x_in_A(A, x)` sẽ được giữ nguyên. Tuy nhiên, ở đây ta sẽ không khởi gán giá trị `x` cũng như `N` nữa, thay vào đó ta sẽ cho `x` là giá trị ngẫu nhiên chạy từ 0 đến 10000 và số phần tử `N` của mảng `A` sẽ tăng từ 10 đến 10000 với bước nhảy là 10. Và thiết lập khoảng giá trị mặc định của từng phần tử trong mảng `A` là thuộc `[0, 10000]`. (File: `bai1_2.py`)

```
1 import numpy as np
2 def search_x_in_A(A, x):
3     save = []
4     for index, value in enumerate(A):
5         if(value == x):
6             save.append(index)
7     if save: return save
8     else: return "Khong co gia tri nao"
9
10 end_range = 10001
11 x = np.random.randint(0,end_range)
12 print("x ngau nhien la:",x)
13
14 for i in range(10, end_range, 10):
15     A = sorted(np.random.randint(0,end_range,i))
16     print("- Mang A co",i,"phan tu.")
17     print(search_x_in_A(A,x))
18
```

“`end_range = 10001`”: Khởi gán biến `end_range` chứa giá trị 10001 để thuận tiện cho việc sửa đổi. Bởi ta coi giá trị `end_range - 1` này là giá trị kết thúc của giá trị ngẫu nhiên `x`, giá trị phần tử thuộc mảng `A` và giá trị kết thúc vòng lặp số phần tử `N` của mảng nên để thuận tiện cho việc sửa đổi cho việc tăng khoảng này lên

thì ta chỉ cần sửa giá trị biến `end_range` thay vì phải sửa từng dòng trên.

“`x = np.random.randint(0, end_range)`”: Sinh ra giá trị ngẫu nhiên nằm trong đoạn từ `[0, end_range - 1]` và gán vào biến `x`.

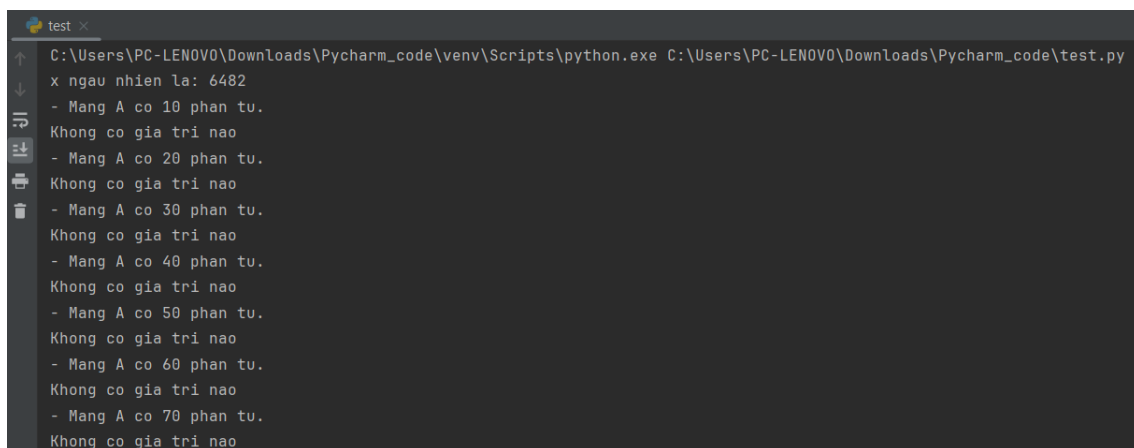
“`print(“x ngẫu nhiên là:”,x)`”: In giá trị `x` để dễ dàng kiểm tra.

“`for i in range(10, end_range, 10):`”: Tạo vòng lặp `for` cho biến `i`, là biến đại diện cho số phần tử `N`, xét tại `i = 10` đến giá trị cuối là `i = end_range - 1` với bước nhảy là 10.

“`A = sorted(np.random.randint(0, end_range, i))`”: Ở mỗi vòng lặp ta sẽ thực hiện lệnh để tạo ngẫu nhiên mảng `A` được sắp xếp tăng dần thông qua hàm `sorted()` và các giá trị trong mảng nằm trong đoạn từ 0 đến `end_range - 1` với thông số cuối là số phần tử của mảng `A` thay đổi qua từng vòng lặp.

Cuối cùng in thông báo số lượng phần tử mảng `A` và thông báo hoặc danh sách các vị trí index có `A[index] = x` hoặc không tồn tại index thỏa điều trên.

Kết quả thu được:



```
test
C:\Users\PC-LENOVO\Downloads\Pycharm_code\venv\Scripts\python.exe C:\Users\PC-LENOVO\Downloads\Pycharm_code\test.py
x ngẫu nhiên là: 6482
- Mảng A có 10 phần tử.
Không có giá trị nào
- Mảng A có 20 phần tử.
Không có giá trị nào
- Mảng A có 30 phần tử.
Không có giá trị nào
- Mảng A có 40 phần tử.
Không có giá trị nào
- Mảng A có 50 phần tử.
Không có giá trị nào
- Mảng A có 60 phần tử.
Không có giá trị nào
- Mảng A có 70 phần tử.
Không có giá trị nào
```



```
test x
↑ [6373]
↓ - Mang A co 9880 phan tu.
[6327]
- Mang A co 9890 phan tu.
- Khong co gia tri nao
- Mang A co 9900 phan tu.
[6406, 6407]
- Mang A co 9910 phan tu.
- Khong co gia tri nao
- Mang A co 9920 phan tu.
[6412, 6413, 6414]
- Mang A co 9930 phan tu.
[6424, 6425, 6426]
- Mang A co 9940 phan tu.
[6437]
- Mang A co 9950 phan tu.
- Khong co gia tri nao
- Mang A co 9960 phan tu.
[6466, 6467]
- Mang A co 9970 phan tu.
- Khong co gia tri nao
- Mang A co 9980 phan tu.
[6424, 6425, 6426]
- Mang A co 9990 phan tu.
- Khong co gia tri nao
- Mang A co 10000 phan tu.
- Khong co gia tri nao

Process finished with exit code 0
```

Bài 2.

2.1. Chương trình với N tăng dần dùng hàm có sẵn.

```
1 import numpy as np
2 k=5
3 for i in range(10, 101, 10):
4     S = sorted(np.random.choice(np.arange(1,1001), size=i, replace=False))
5     print("- Tap S co",i,"phan tu.")
6     print(S)
7     print("Phan tu thu",k,"cua tap S la:",S[k-1])
8
```

(File: bai2_1.py)

“import numpy as np”: Gọi thư viện numpy để thao tác với mảng trong đoạn mã của chương trình và gọi tắt là “np”, điều này cho phép gọi các hàm và đối tượng từ thư viện numpy bằng cách sử dụng tiền tố “np”.

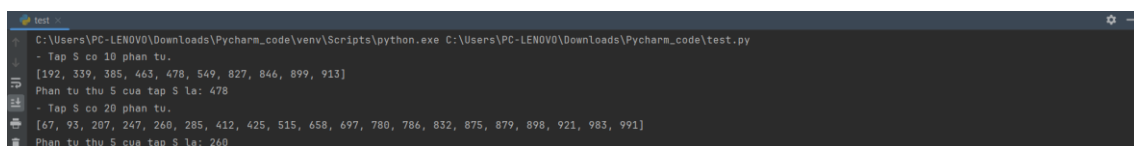
“k=5”: Khởi tạo giá trị ban đầu cho k là 5. Phần tử nhỏ nhất thứ k sẽ mang giá trị của mảng A tại index k - 1.

“for i in range(10, 101, 10):” Tạo vòng lặp for cho biến i, là đại diện cho số phần tử N, mang giá trị từ 10 đến 100 với bước nhảy là 10 như dữ liệu test của bài toán.

“S = sorted(np.random.randint(0, 1001, i))”: Ở mỗi vòng lặp ta sẽ thực hiện lệnh để tạo ngẫu nhiên tập S được sắp xếp tăng dần thông qua hàm sorted() và các giá trị trong tập nằm trong đoạn từ 0 đến 1000 với thông số cuối là số phần tử của tập S thay đổi qua từng vòng lặp.

Và in ra thông báo về từng vòng lặp như số phần tử hiện có của tập S “print(- Tap S có “i,”phần tử.)”, vì số lượng phần tử không quá nhiều nên ta cũng in tập S ra để dễ quan sát “print(S)” và dòng print cuối cùng là in phần tử nhỏ nhất thứ k của tập hợp S chính là giá trị của mảng S tại index k - 1.

Kết quả.



```
test
C:\Users\PC-LENOVO\Downloads\Pycharm_code\venv\Scripts\python.exe C:\Users\PC-LENOVO\Downloads\Pycharm_code\test.py
- Tap S có 10 phần tử.
[192, 339, 385, 463, 478, 549, 827, 846, 899, 913]
Phần tử thứ 5 của tập S là: 478
- Tap S có 20 phần tử.
[67, 93, 207, 247, 260, 285, 412, 425, 515, 658, 697, 780, 786, 832, 875, 879, 898, 921, 983, 991]
Phần tử thứ 5 của tập S là: 260
```

```
test -
- Tap S có 30 phần tử.
[60, 81, 124, 132, 140, 170, 173, 194, 199, 210, 225, 238, 256, 271, 406, 419, 420, 422, 530, 633, 683, 720, 800, 826, 849, 903, 929, 966, 980, 999]
Phan tu thu 5 của tap S la: 140
- Tap S có 40 phần tử.
[26, 60, 67, 86, 87, 117, 134, 298, 319, 344, 372, 388, 497, 516, 531, 598, 616, 621, 632, 687, 705, 721, 728, 735, 757, 760, 883, 821, 848, 856, 868, 881, 890, 908, 910, 912, 913]
Phan tu thu 5 của tap S la: 87
- Tap S có 50 phần tử.
[5, 20, 23, 42, 70, 99, 100, 105, 116, 128, 130, 136, 147, 157, 201, 244, 247, 252, 267, 297, 330, 351, 355, 380, 394, 401, 404, 438, 448, 455, 462, 498, 525, 595, 598, 599, 600]
Phan tu thu 5 của tap S la: 70
- Tap S có 60 phần tử.
[12, 25, 40, 43, 46, 47, 84, 86, 101, 114, 124, 142, 157, 172, 175, 181, 216, 221, 223, 226, 235, 248, 266, 267, 300, 312, 323, 324, 347, 354, 362, 410, 426, 437, 456, 459, 461]
Phan tu thu 5 của tap S la: 46
- Tap S có 70 phần tử.
[1, 2, 14, 22, 24, 49, 51, 55, 70, 77, 92, 118, 134, 137, 153, 157, 161, 162, 174, 175, 189, 196, 216, 226, 248, 297, 311, 312, 332, 361, 365, 368, 420, 453, 464, 466, 493, 496]
Phan tu thu 5 của tap S la: 24
- Tap S có 80 phần tử.
[2, 5, 29, 30, 60, 67, 77, 92, 97, 112, 119, 120, 132, 133, 145, 173, 179, 183, 189, 193, 195, 206, 214, 215, 228, 251, 279, 286, 297, 310, 319, 324, 334, 360, 364, 368, 388, 420]
Phan tu thu 5 của tap S la: 60
- Tap S có 90 phần tử.
[41, 47, 77, 98, 113, 122, 125, 163, 169, 176, 212, 223, 244, 255, 259, 266, 267, 291, 306, 308, 323, 328, 338, 348, 350, 357, 362, 375, 383, 396, 397, 399, 412, 426, 429, 474, 496]
Phan tu thu 5 của tap S la: 113
- Tap S có 100 phần tử.
[7, 18, 36, 37, 39, 48, 53, 60, 70, 74, 83, 90, 94, 114, 120, 143, 155, 168, 170, 172, 174, 213, 224, 225, 235, 240, 261, 266, 272, 273, 279, 280, 282, 283, 298, 304, 309, 313, 314]
Phan tu thu 5 của tap S la: 39
Process finished with exit code 0
```

2.2. Chương trình với N tăng dần dùng hàm tự định nghĩa.

Bởi cần đếm số phép so sánh và phép gán cho từ việc tính thời gian trung bình để $S(i) \neq S(j) \forall i \neq j$, mà nếu dùng hàm có sẵn sẽ tùy chương trình, phần mềm mà hàm tự định nghĩa sẽ có sự khác nhau hoặc ẩn hàm vì tính bảo mật. Nên ta sẽ tự định nghĩa hàm để tạo tập S ngẫu nhiên có N phần tử khác nhau và sắp xếp tăng dần bằng selection sort. (File: bai2_2.py)

```
1 import numpy as np
2 k=5
3
4 def selectionSort(arr):
5     for i in range(len(arr)):
6         min_idx = i
7         for j in range(i+1, len(arr)):
8             if arr[min_idx]>arr[j]:
9                 min_idx=j
10        arr[i], arr[min_idx] = arr[min_idx], arr[i]
11    return arr
```

“import numpy as np”: Gọi thư viện numpy để thao tác với mảng trong đoạn mã của chương trình và gọi tắt là “np”, điều này cho phép gọi các hàm và đối tượng từ thư viện numpy bằng cách sử dụng tiền tố “np”.

“k=5”: Khởi tạo giá trị ban đầu cho k là 5. Phần tử nhỏ nhất thứ k sẽ mang giá trị của tập S tại index k - 1.

“def selectionSort(arr):” Định nghĩa hàm selection sort với đối số truyền vào là 1 mảng cần sắp xếp theo thứ tự tăng dần.

“for i in range(len(arr)):”: Tạo vòng lặp for để duyệt các phần tử trong mảng S. Lúc này, i chạy từ 0 đến len(arr)-1. Với từng vị trí đang xét i đó, ta khởi tạo biến ‘min_idx = i’ để min_idx chứa giá trị index nhỏ nhất tạm thời là i, rồi duyệt từ vị trí sau i đến cuối mảng S ‘for j in range(i+1, len(arr))’. Nếu trong quá trình duyệt mà tồn tại phần tử nào mà giá trị tại đó nhỏ hơn giá trị tại index min_idx ‘if arr[min_idx]>arr[j]’ thì ta cập nhật lại min_idx ‘min_idx=j’. Kết thúc vòng lặp j, ta sẽ tìm ra được vị trí nhỏ nhất từ index i+1 đến cuối mảng, lúc này ta đổi giá trị tại index thứ i cho min_idx và ngược lại. Hàm trả về mảng đã được sắp xếp ‘return arr’.

```
13 for i in range(10, 101, 10):
14     S = []
15     for m in range(0, i):
16         count_N=m
17         check_scanned=False
18         while(check_scanned == False):
19             check_scanned=True
20             j = np.random.randint(1, 1001)
21             for t in range(0, count_N):
22                 if (S[t] == j):
23                     check_scanned = False
24                     break
25             S.append(j)
26     print("- Tap S co",i,"phan tu.")
27     print("+ Mang S ban dau:\n",S)
28     print("+ Sau khi sort S:\n",selectionSort(S))
29     print("+ Phan tu thu",k,"cua tap S la:",S[k-1],"\n")
30
```

“for i in range(10, 101, 10):”: Tạo vòng lặp for với giá trị i duyệt từ 10 đến 100 và bước nhảy là 10, đây là biến tượng trưng cho N trong bài.

“S = []”: Tạo một list S rỗng, đại diện cho tập S trong bài. Ở các bước tiếp theo, ta sẽ xét một phần tử được tạo ngẫu nhiên bằng hàm random() rồi so sánh với các phần tử trước đó, nếu như

phần tử đó chưa xuất hiện trong S thì ta sẽ dùng phương thức `append()` để lưu lại.

“for m in range(0,i):”: Tạo vòng lặp m chạy từ 0 đến i-1, đây là vòng lặp được hiểu là duyệt từng vị trí của mảng S ta hướng đến là có N phần tử, tức là i ở vòng lặp này.

Khởi tạo biến `count_N=m` để lưu index tại vị trí đang xét, từ đó ta sẽ tạo vòng lặp xét từ đầu đến vị trí trước index `count_N` đó.

Khởi tạo biến `check_scanned=False` để lưu biến kiểu bool là False, chủ yếu là để điều kiện cần ban đầu để bước vào vòng lặp while.

“while(`check_scanned == False`):” Tạo vòng lặp while để kiểm tra giá trị `check_scanned` có là False hay không? Nếu là False tức mang ý nghĩa là trong quá trình xét từ vị trí đầu đến index `count_N` thì có 1 phần tử đã được sinh ra và trùng với giá trị vừa được tạo ra, do đó tiếp tục vòng lặp while để tạo ra giá trị khác với các giá trị trước đó.

Trong while ta lại gán `check_scanned` ban đầu bằng True, và tạo giá trị ngẫu nhiên j là số nguyên dương trong đoạn từ 1 đến 1000. Tạo tiếp vòng lặp for t như ý tưởng đã nêu là xét từ index 0 đến index hiện tại là `count_N=m`. Nếu giá trị j đó đã xuất hiện ‘if (`S[t] == j`):’ thì ta gán `check_scanned = False` và break vòng lặp đó.

Kết thúc vòng lặp while, ta thu được giá trị ngẫu nhiên tại vị trí thứ m, thỏa giá trị từ index 0 đến index m là khác nhau. Ta dùng phương thức `append()` để thêm giá trị đó vào S.

Cuối cùng ta in lần lượt số lượng phần tử tập S, S được tạo ra trước và sau khi sắp xếp. Và rồi in giá trị thứ k của S.

Kết quả thu được:

```
test
C:\Users\PC-LENOVO\Downloads\Pycharm_code\venv\Scripts\python.exe C:\Users\PC-LENOVO\Downloads\Pycharm_code\test.py
- Tap S có 10 phần tử.
+ Mang S ban đầu:
[56, 554, 999, 44, 166, 334, 746, 320, 68, 362]
+ Sau khi sort S:
[44, 56, 68, 166, 320, 334, 362, 554, 746, 999]
+ Phần tử thứ 5 của tap S là: 320

- Tap S có 20 phần tử.
+ Mang S ban đầu:
[709, 902, 710, 941, 275, 821, 633, 755, 886, 251, 665, 989, 348, 968, 896, 288, 973, 133, 457, 632]
+ Sau khi sort S:
[133, 251, 275, 288, 348, 457, 632, 633, 665, 709, 710, 755, 821, 886, 896, 902, 941, 968, 973, 989]
+ Phần tử thứ 5 của tap S là: 348

- Tap S có 30 phần tử.
+ Mang S ban đầu:
[577, 539, 574, 851, 889, 940, 282, 658, 319, 848, 732, 75, 600, 218, 522, 26, 836, 884, 529, 672, 359, 930, 950, 519, 355, 61, 456, 247, 542, 149]
+ Sau khi sort S:
[26, 61, 75, 149, 218, 247, 282, 319, 355, 359, 456, 519, 522, 529, 539, 542, 574, 577, 600, 658, 672, 732, 836, 848, 851, 884, 889, 930, 940, 950]
+ Phần tử thứ 5 của tap S là: 218

- Tap S có 40 phần tử.
+ Mang S ban đầu:
[632, 629, 388, 483, 894, 618, 213, 955, 637, 510, 783, 900, 931, 18, 989, 12, 824, 889, 335, 158, 137, 999, 273, 123, 856, 857, 346, 299, 309, 88, 617, 460, 70, 160, 578, 480, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999]
+ Sau khi sort S:
[1, 12, 18, 70, 88, 123, 137, 158, 160, 213, 232, 270, 273, 299, 309, 335, 346, 388, 460, 480, 483, 510, 578, 617, 618, 629, 632, 637, 783, 805, 824, 856, 857, 889, 894, 900, 930, 940, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999]
+ Phần tử thứ 5 của tap S là: 88
```

```
test
- Tap S có 50 phần tử.
+ Mang S ban đầu:
[434, 375, 308, 523, 894, 258, 417, 919, 967, 220, 386, 886, 26, 825, 377, 37, 675, 791, 592, 614, 255, 159, 138, 879, 349, 1, 973, 604, 351, 552, 234, 680, 398, 820, 298, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500]
+ Sau khi sort S:
[1, 17, 26, 37, 138, 159, 178, 220, 234, 255, 257, 258, 290, 298, 308, 349, 351, 365, 375, 377, 385, 386, 398, 405, 417, 423, 434, 470, 523, 552, 592, 604, 614, 649, 675, 680, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999]
+ Phần tử thứ 5 của tap S là: 138

- Tap S có 60 phần tử.
+ Mang S ban đầu:
[661, 710, 514, 84, 221, 329, 137, 180, 859, 678, 624, 593, 314, 561, 639, 874, 757, 900, 890, 388, 979, 541, 786, 908, 103, 377, 451, 160, 23, 722, 302, 125, 354, 873, 335, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999]
+ Sau khi sort S:
[23, 84, 103, 108, 125, 137, 160, 180, 186, 209, 221, 242, 293, 302, 314, 329, 335, 339, 344, 354, 363, 377, 388, 451, 501, 514, 541, 546, 561, 593, 623, 624, 629, 639, 652, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999]
+ Phần tử thứ 5 của tap S là: 125

- Tap S có 70 phần tử.
+ Mang S ban đầu:
[613, 50, 661, 542, 14, 564, 413, 432, 722, 972, 362, 182, 89, 645, 308, 578, 676, 669, 319, 329, 445, 540, 519, 31, 67, 897, 820, 796, 969, 307, 777, 466, 17, 671, 810, 713, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]
+ Sau khi sort S:
[14, 17, 21, 31, 50, 51, 54, 67, 89, 131, 142, 153, 154, 182, 274, 307, 308, 319, 325, 329, 361, 362, 413, 429, 432, 434, 439, 441, 445, 458, 466, 472, 494, 509, 515, 519, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999]
+ Phần tử thứ 5 của tap S là: 50
```

```
test
- Tap S có 80 phần tử.
+ Mang S ban đầu:
[975, 868, 286, 337, 166, 400, 910, 852, 409, 95, 478, 433, 529, 331, 387, 447, 953, 473, 738, 9, 235, 410, 965, 602, 278, 483, 742, 257, 625, 849, 308, 957, 156, 767, 682, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999]
+ Sau khi sort S:
[9, 12, 47, 62, 91, 95, 115, 145, 152, 155, 156, 166, 178, 186, 202, 213, 234, 235, 257, 278, 286, 288, 301, 308, 315, 331, 337, 342, 346, 387, 400, 409, 410, 414, 433, 445, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500]
+ Phần tử thứ 5 của tap S là: 91

- Tap S có 90 phần tử.
+ Mang S ban đầu:
[635, 687, 720, 1000, 671, 57, 840, 992, 799, 86, 2, 630, 449, 364, 714, 83, 317, 984, 998, 826, 78, 567, 997, 661, 454, 144, 467, 190, 424, 260, 929, 126, 978, 514, 855, 326, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]
+ Sau khi sort S:
[2, 4, 19, 24, 42, 49, 55, 57, 65, 73, 78, 83, 84, 86, 87, 102, 118, 121, 126, 144, 163, 173, 190, 229, 260, 270, 306, 310, 313, 317, 326, 364, 373, 378, 392, 394, 419, 424, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452,
```

```

1  import numpy as np
2  k=5
3  S = []
4  end = 101
5
6  count_sosanh = 0
7  count_gan = 1
8
9
10 def selectionSort(arr, count_sosanh, count_gan):
11     for i in range(len(arr)):
12         min_idx = i
13         for j in range(i+1, len(arr)):
14             if arr[min_idx]>arr[j]:
15                 min_idx=j
16                 count_gan+=1
17                 count_sosanh+=1
18             arr[i], arr[min_idx] = arr[min_idx], arr[i]
19             count_gan+=3
20     return arr
21

```

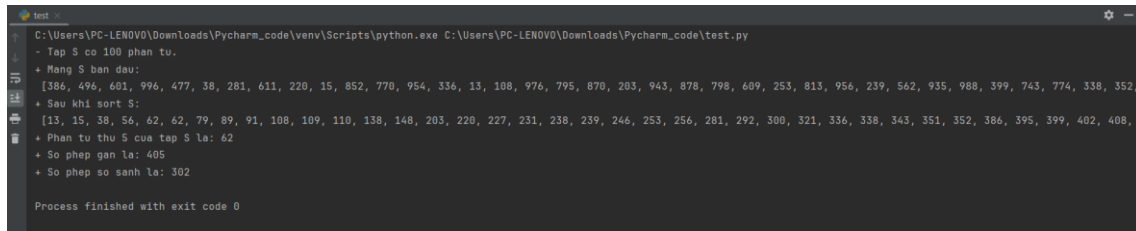
```

22 for m in range(0, end):
23     count_N=m
24     check_scanned=False
25
26     count_gan+=2
27
28     while(check_scanned==False):
29         count_sosanh+=1
30
31         check_scanned=True
32         j = np.random.randint(1, 1001)
33
34         count_gan+=2
35         for t in range(0, count_N):
36             if (S[t] == j):
37                 check_scanned = False
38                 count_gan += 1
39                 count_sosanh += 1
40                 break
41
42         count_sosanh+=1
43         S.append(j)
44     print("- Tap S co",end-1,"phan tu.")
45     print("+ Mang S ban dau:\n",S)
46     print("+ Sau khi sort S:\n",selectionSort(S, count_sosanh, count_gan))
47     print("+ Phan tu thu",k,"cua tap S la:",S[k-1])
48     print("+ So phiep gan la:",count_gan)
49     print("+ So phiep so sanh la:",count_sosanh)
50

```

Sẽ có điều chỉnh đầu vào ở function `selectionSort()` là 3 đối số truyền vào, vì để gọn và dễ hiểu nên ta truyền các số vào để đếm luôn thay vì viết thêm đoạn mã cộng ngay sau khi gọi function. Và hàm vẫn như bài trên là sẽ trả về mảng đã được sắp xếp.

Kết quả thu được:



```
C:\Users\PC-LENOVO\Downloads\Pycharm_code\venv\Scripts\python.exe C:\Users\PC-LENOVO\Downloads\Pycharm_code\test.py
- Tập S có 100 phần tử.
+ Mảng S ban đầu:
[386, 496, 601, 996, 477, 38, 281, 611, 220, 15, 852, 770, 954, 336, 13, 108, 976, 795, 870, 203, 943, 878, 798, 609, 253, 813, 956, 239, 562, 935, 988, 399, 743, 774, 338, 352,
+ Sau khi sort S:
[13, 15, 38, 56, 62, 62, 79, 89, 91, 108, 109, 110, 138, 148, 203, 220, 227, 231, 238, 239, 246, 253, 256, 281, 292, 300, 321, 336, 338, 343, 351, 352, 386, 395, 399, 402, 408,
+ Phần tử thứ 5 của tập S là: 62
+ Số phép gán là: 405
+ Số phép so sánh là: 302

Process finished with exit code 0
```