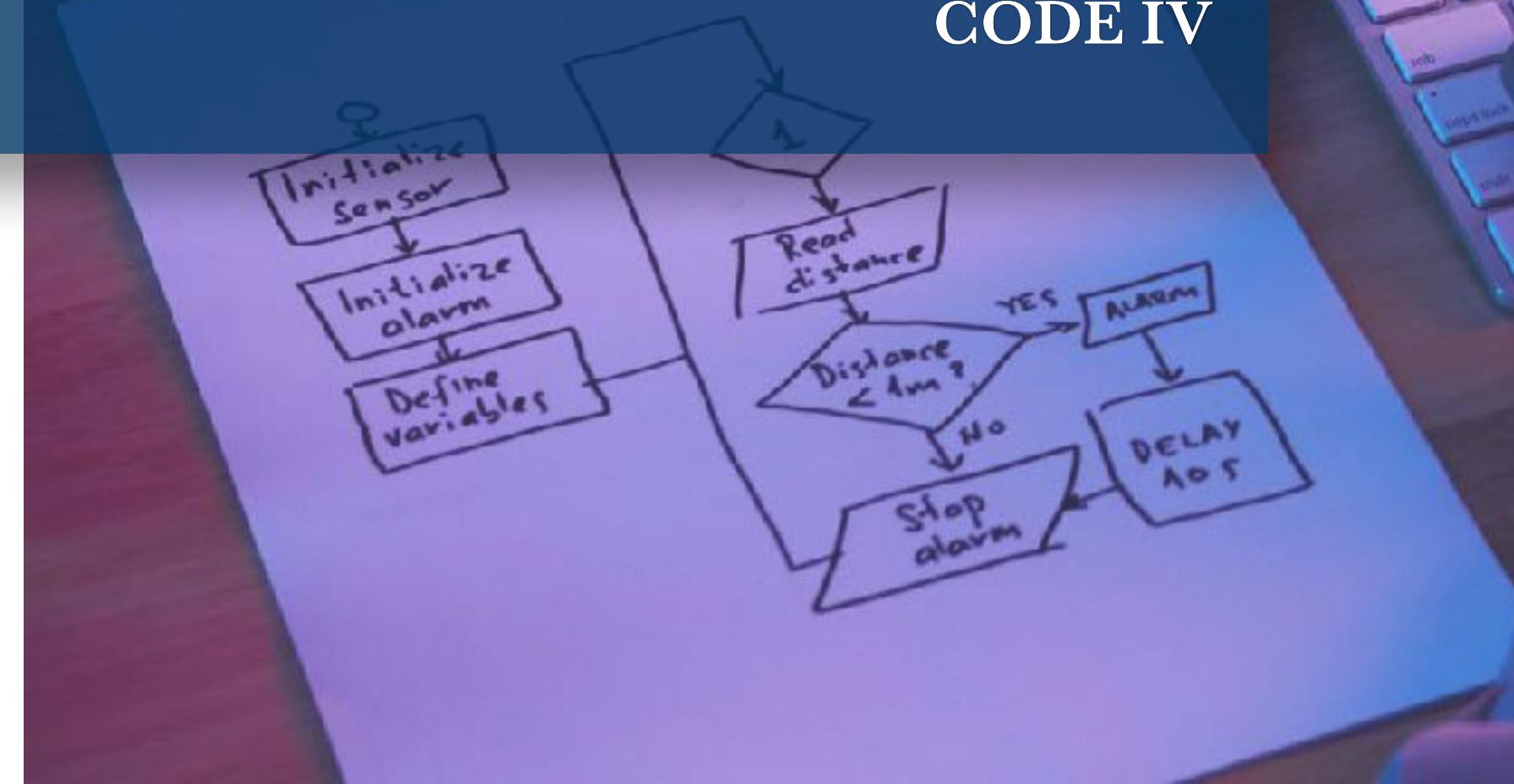


DATA STRUCTURES AND ALGORITHMS

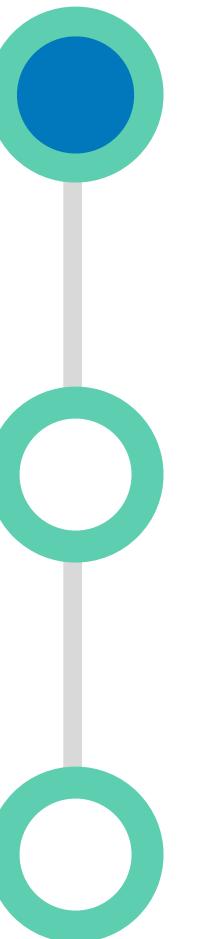
IT003.N27.1
GROUP 4
CODE IV



SƠ LƯỢC NỘI DUNG CHÍNH

LÊ MINH QUÂN

22521181



Sắp Xếp, Cây Nhị Phân Tùm Kiếm
Tùm Kiếm Nhị Phân

LẠI QUAN THIÊN

22521385

Danh Sách Liên Kết
Bảng Băm

TRẦN THẾ HỮU PHÚC

22521143

Cây Nhị Phân

Group 4



PHẦN I

SẮP XẾP TÌM KIẾM NHỊ PHÂN CÂY NHỊ PHÂN

Đề 10: Câu 1, 2 - Đề 11: câu 2

XEM THÊM

LÊ MINH QUÂN

22521181



SẮP XẾP

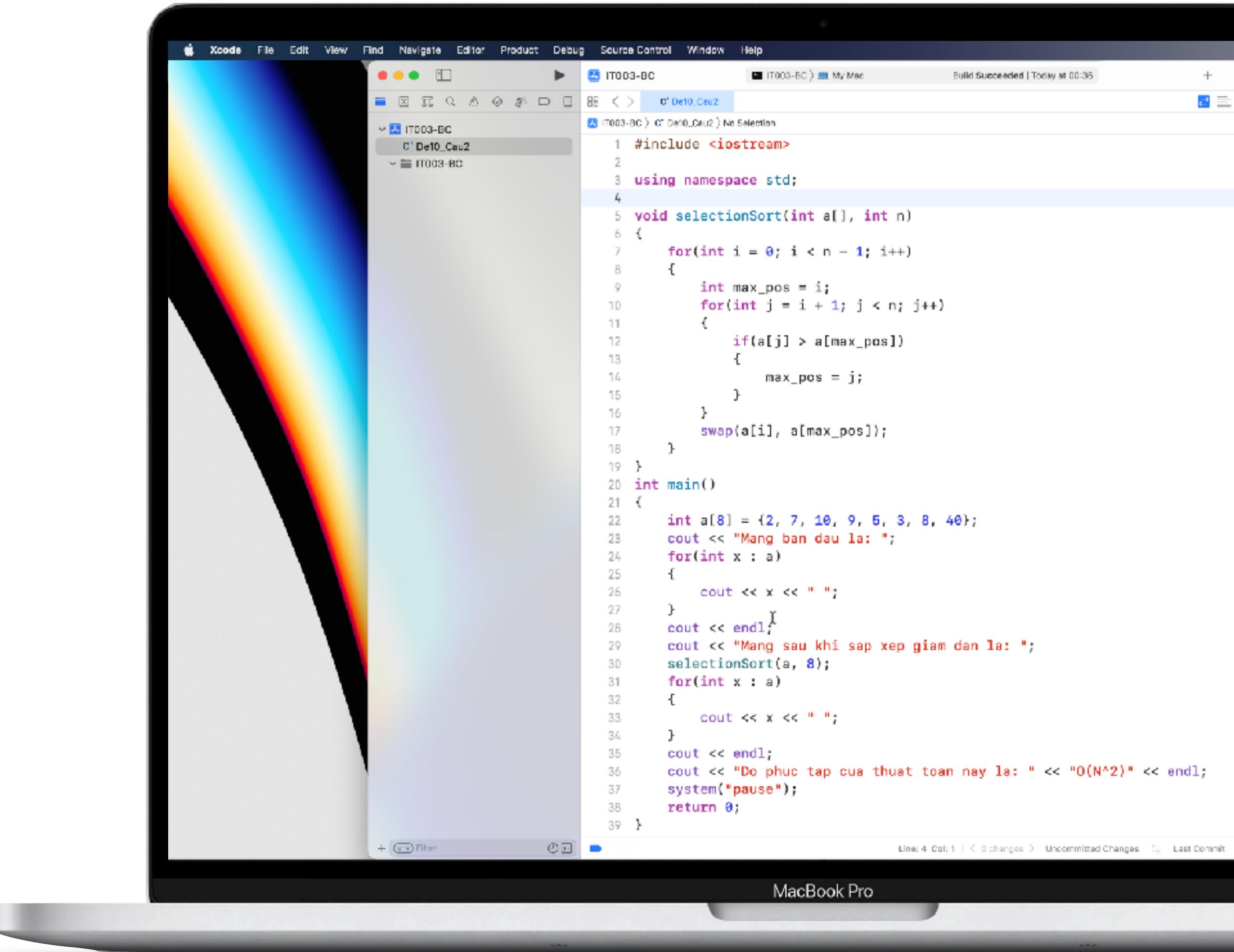
Đề 10: Câu 2

ĐỀ BÀI

Đề 10: Câu 2

- a.** Anh/chị hãy viết hàm cài đặt thuật toán chọn trực tiếp (**Selection Sort**) để sắp xếp một mảng các số nguyên có N phần tử **theo chiều giảm dần** bằng ngôn ngữ C/C++, cho biết độ phức tạp của giải thuật.
- b.** Trình bày các bước (vẽ từng bước) áp dụng thuật toán ở câu 2.a để **sắp xếp mảng số nguyên {2, 7, 10, 9, 5, 3, 8, 40} giảm dần**.

a. Anh/chị hãy viết hàm cài đặt thuật toán chọn trực tiếp (**Selection Sort**) để sắp xếp một mảng các số nguyên có N phần tử **theo chiều giảm dần** bằng ngôn ngữ C/C++, cho biết độ phức tạp của giải thuật.

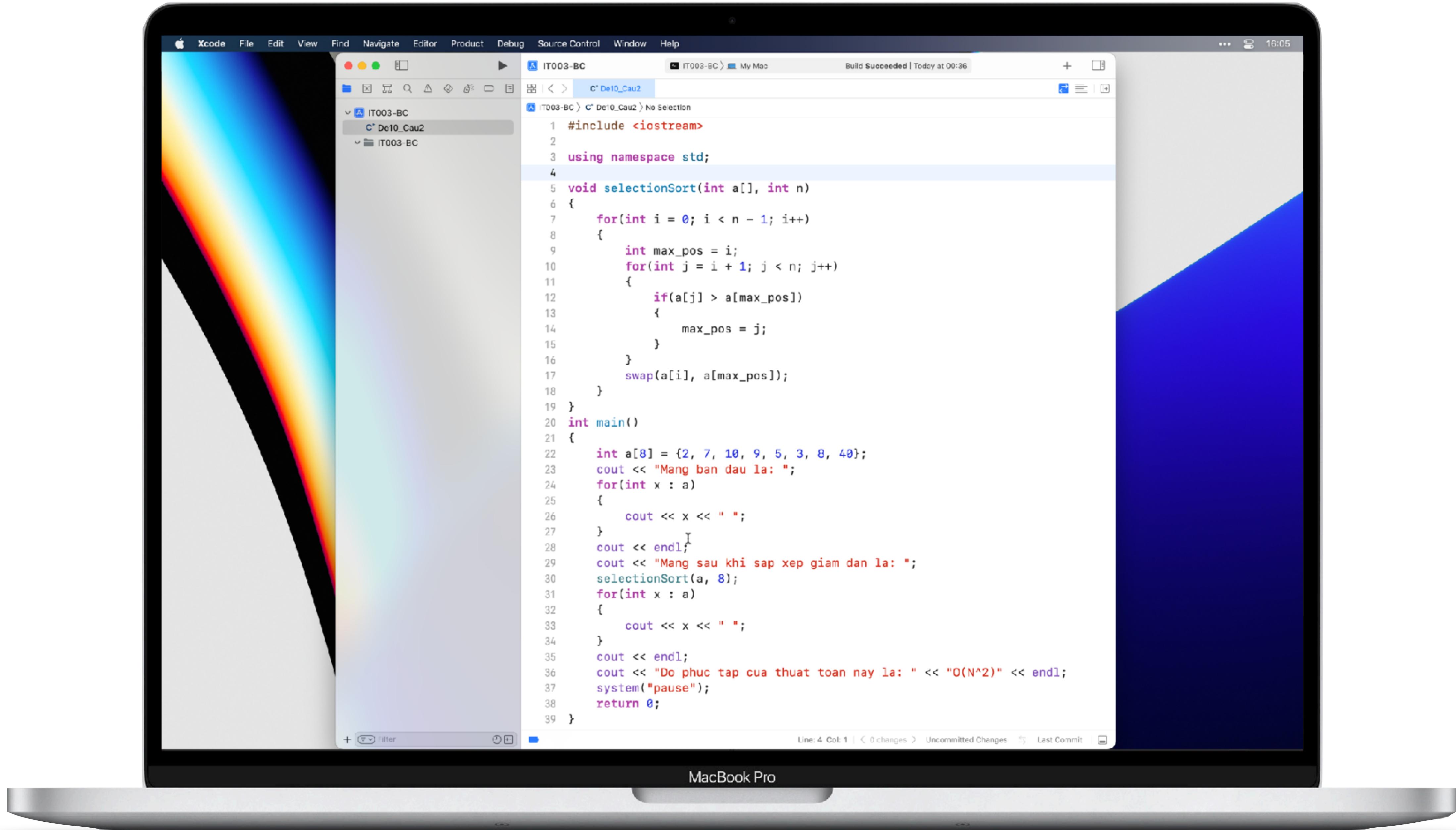


The image shows a MacBook Pro with its screen displaying the Xcode IDE. The project 'IT003-BC' is open, and the file 'C'De10_Cau2.cpp' is selected. The code implements the Selection Sort algorithm in C++ to sort an array of integers in descending order. The code includes a main function that prints the original and sorted arrays, and a note about the time complexity of the algorithm.

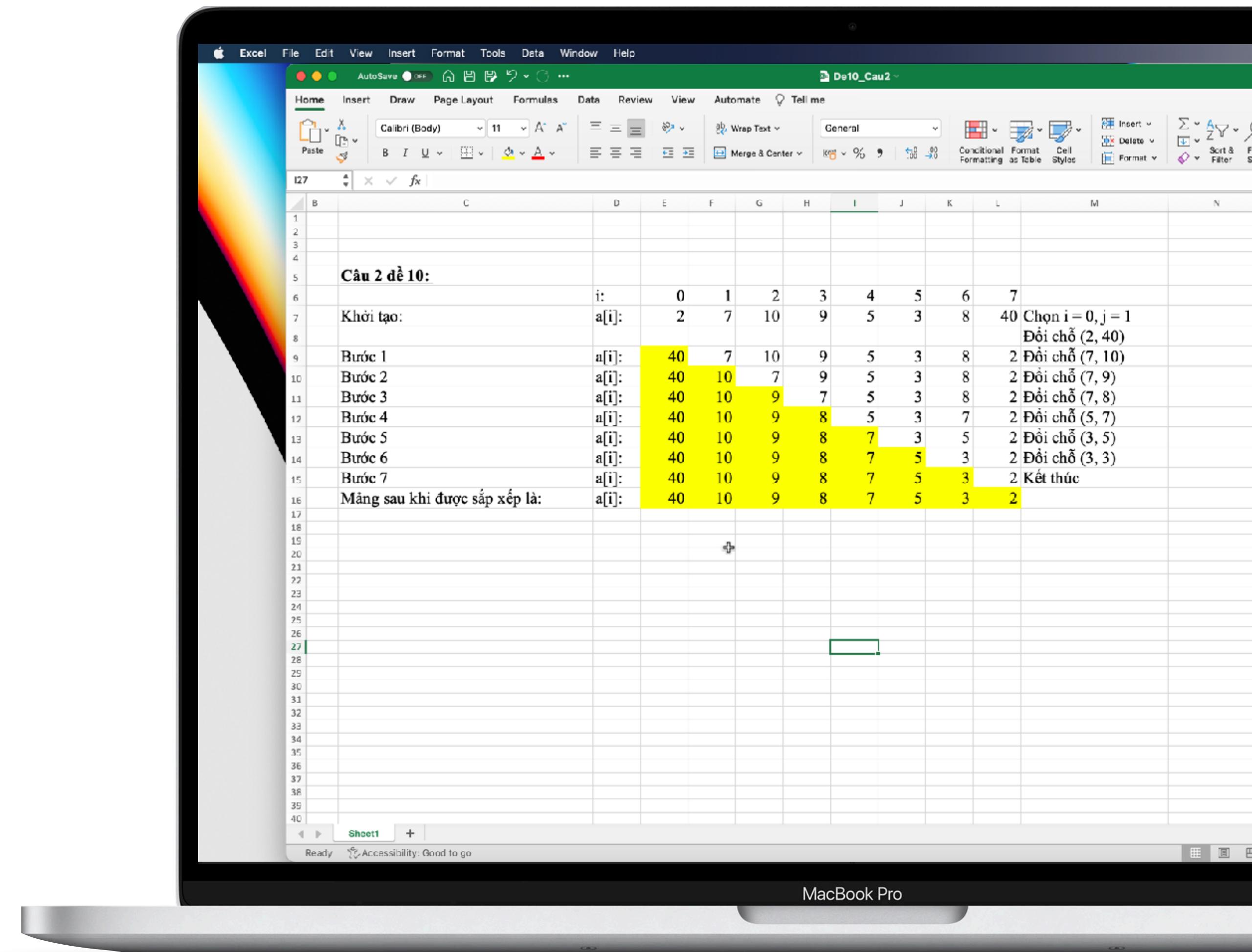
```
#include <iostream>
using namespace std;

void selectionSort(int a[], int n)
{
    for(int i = 0; i < n - 1; i++)
    {
        int max_pos = i;
        for(int j = i + 1; j < n; j++)
        {
            if(a[j] > a[max_pos])
            {
                max_pos = j;
            }
        }
        swap(a[i], a[max_pos]);
    }
}

int main()
{
    int a[8] = {2, 7, 10, 9, 5, 3, 8, 40};
    cout << "Mang ban dau la: ";
    for(int x : a)
    {
        cout << x << " ";
    }
    cout << endl;
    cout << "Mang sau khi sap xep giam dan la: ";
    selectionSort(a, 8);
    for(int x : a)
    {
        cout << x << " ";
    }
    cout << endl;
    cout << "Do phuc tap cua thuat toan nay la: " << "O(N^2)" << endl;
    system("pause");
    return 0;
}
```



b. Trình bày các bước (vẽ từng bước) áp dụng thuật toán ở câu 2.a để **sắp xếp mảng số nguyên $\{2, 7, 10, 9, 5, 3, 8, 40\}$ giảm dần.**



The image shows a MacBook Pro laptop screen displaying an Excel spreadsheet. The spreadsheet is titled "De10_Cau2" and contains a table illustrating the bubble sort algorithm. The table has columns for index "i:" and array elements "a[i]:". The steps are as follows:

	i:	0	1	2	3	4	5	6	7	
Khởi tạo:	a[i]:	2	7	10	9	5	3	8	40	Chọn i=0, j=1
Bước 1	a[i]:	40	7	10	9	5	3	8		Đổi chỗ (2, 40)
Bước 2	a[i]:	40	10	7	9	5	3	8		Đổi chỗ (7, 10)
Bước 3	a[i]:	40	10	9	7	5	3	8		Đổi chỗ (7, 9)
Bước 4	a[i]:	40	10	9	8	5	3	7		Đổi chỗ (9, 8)
Bước 5	a[i]:	40	10	9	8	7	3	5		Đổi chỗ (8, 7)
Bước 6	a[i]:	40	10	9	8	7	5	3		Đổi chỗ (7, 5)
Bước 7	a[i]:	40	10	9	8	7	5	3		Đổi chỗ (5, 3)
Mảng sau khi được sắp xếp là:	a[i]:	40	10	9	8	7	5	3	2	Kết thúc

MacBook Pro

MINH HOẠ THUẬT TOÁN

SELECTION SORT

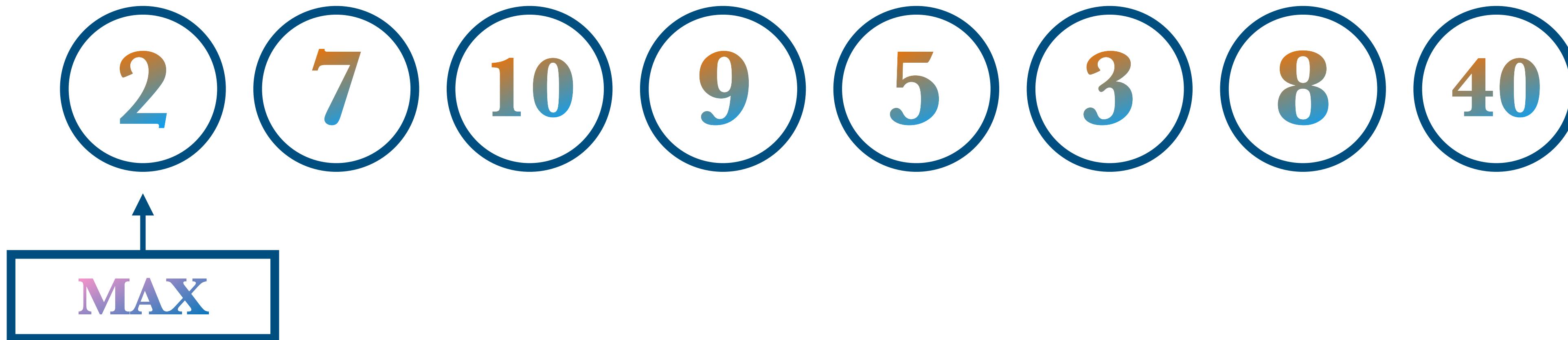
GIẢM DẦN



MINH HOẠ THUẬT TOÁN

SELECTION SORT

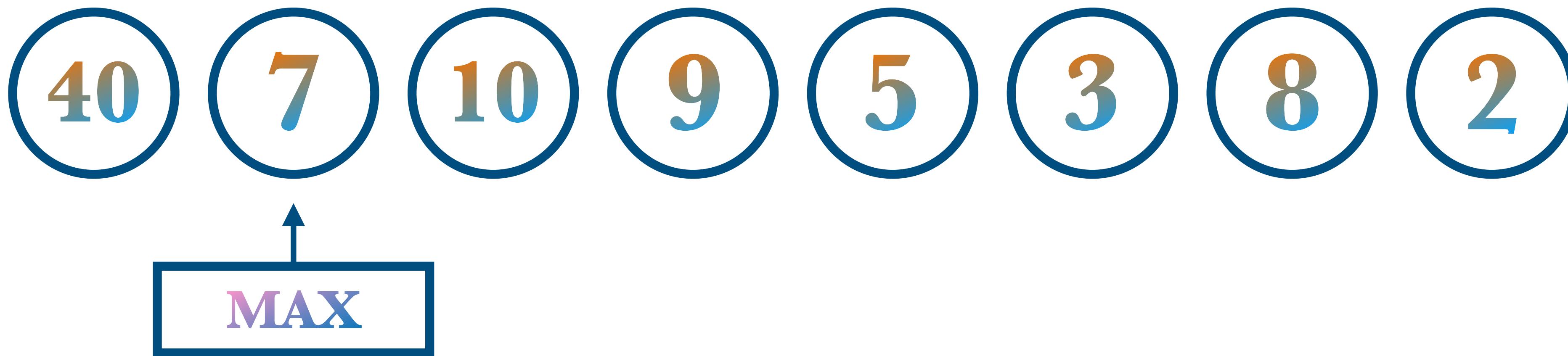
GIẢM DẦN



MINH HOẠ THUẬT TOÁN

SELECTION SORT

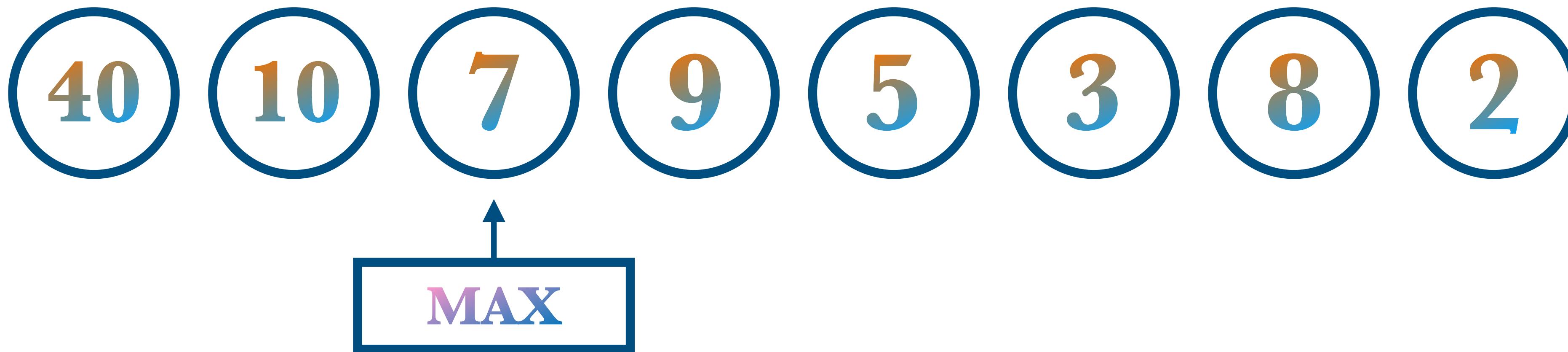
GIẢM DẦN



MINH HOẠ THUẬT TOÁN

SELECTION SORT

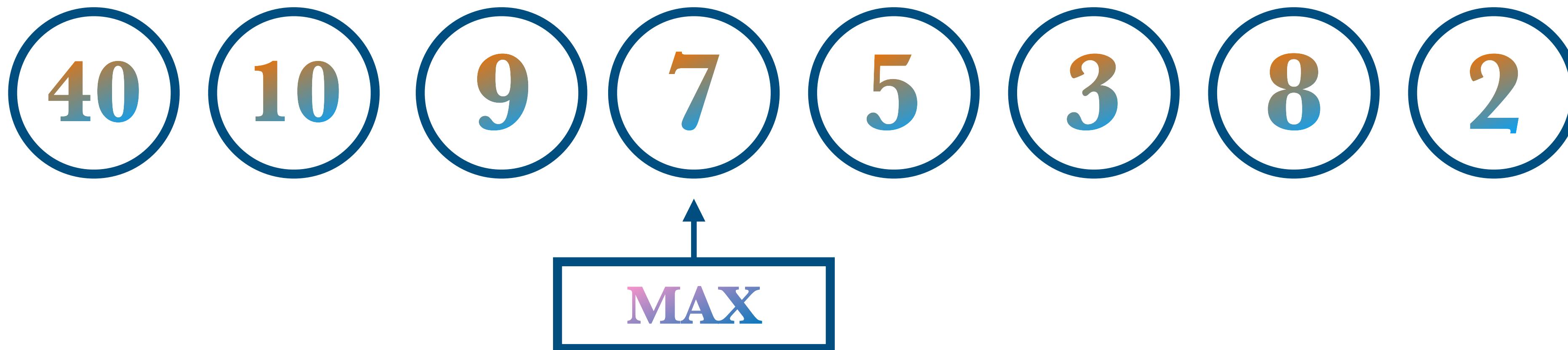
GIẢM DẦN



MINH HOẠ THUẬT TOÁN

SELECTION SORT

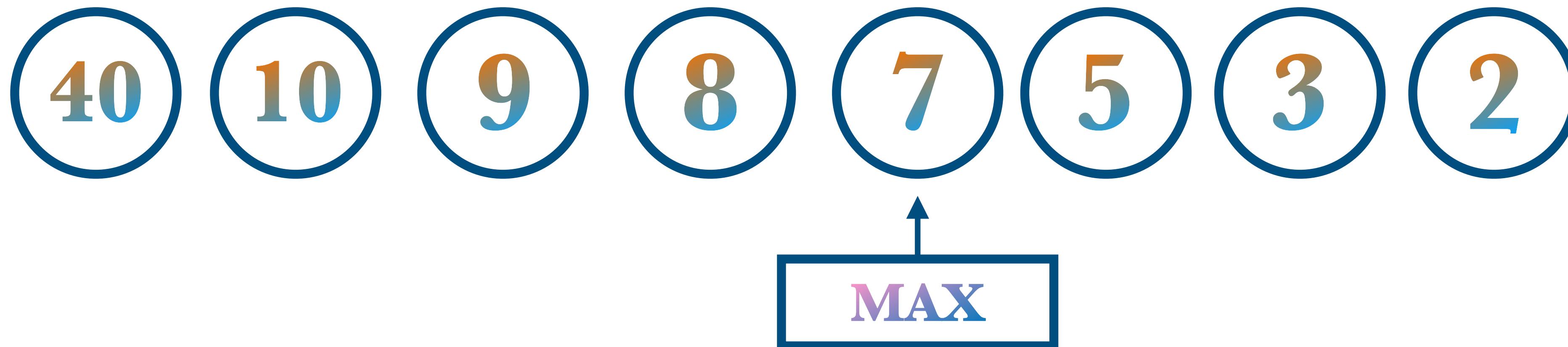
GIẢM DẦN



MINH HOẠ THUẬT TOÁN

SELECTION SORT

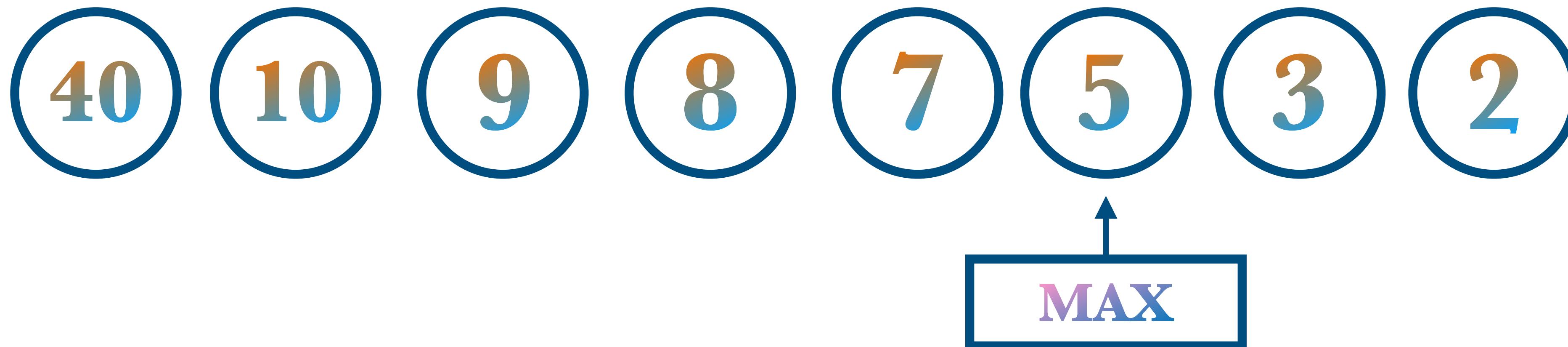
GIẢM DẦN



MINH HOẠ THUẬT TOÁN

SELECTION SORT

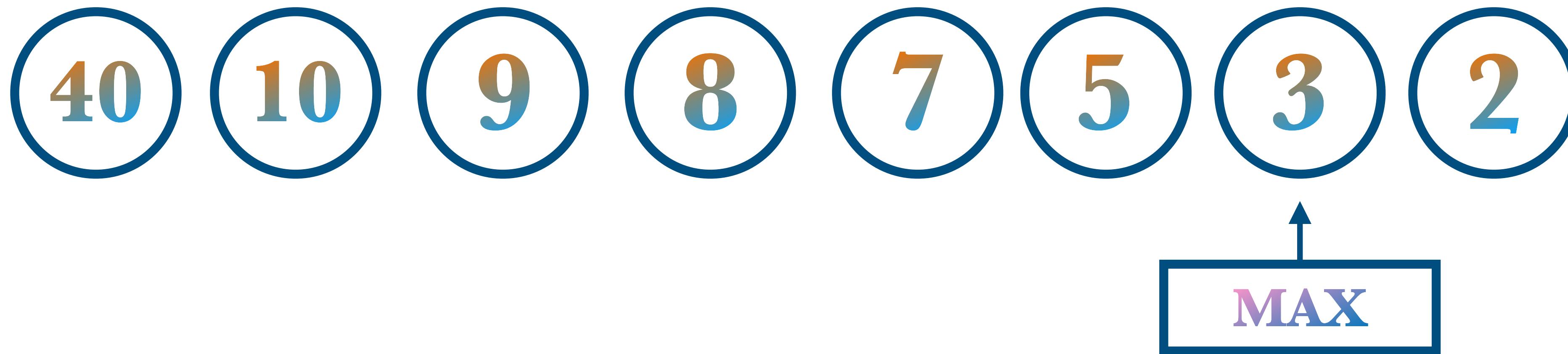
GIẢM DẦN



MINH HOẠ THUẬT TOÁN

SELECTION SORT

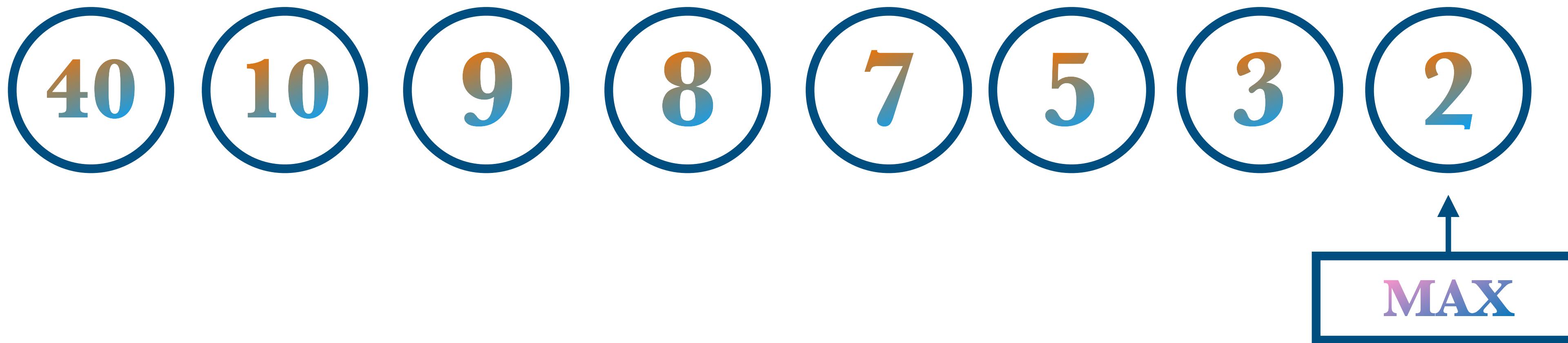
GIẢM DẦN



MINH HOẠ THUẬT TOÁN

SELECTION SORT

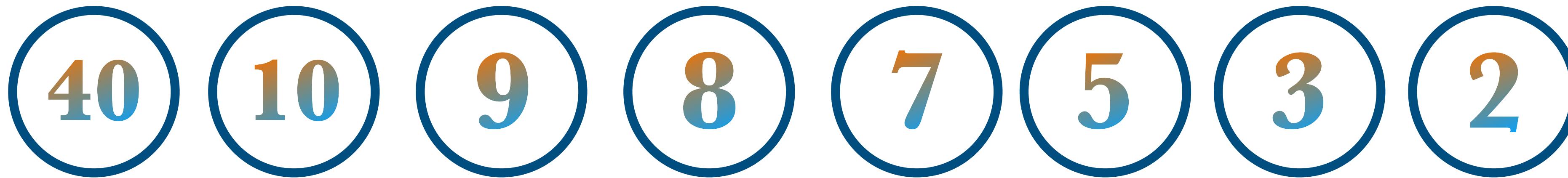
GIẢM DẦN



MINH HOẠ THUẬT TOÁN

SELECTION SORT

GIẢM DẦN



TÌM KIẾM NHỊ PHÂN

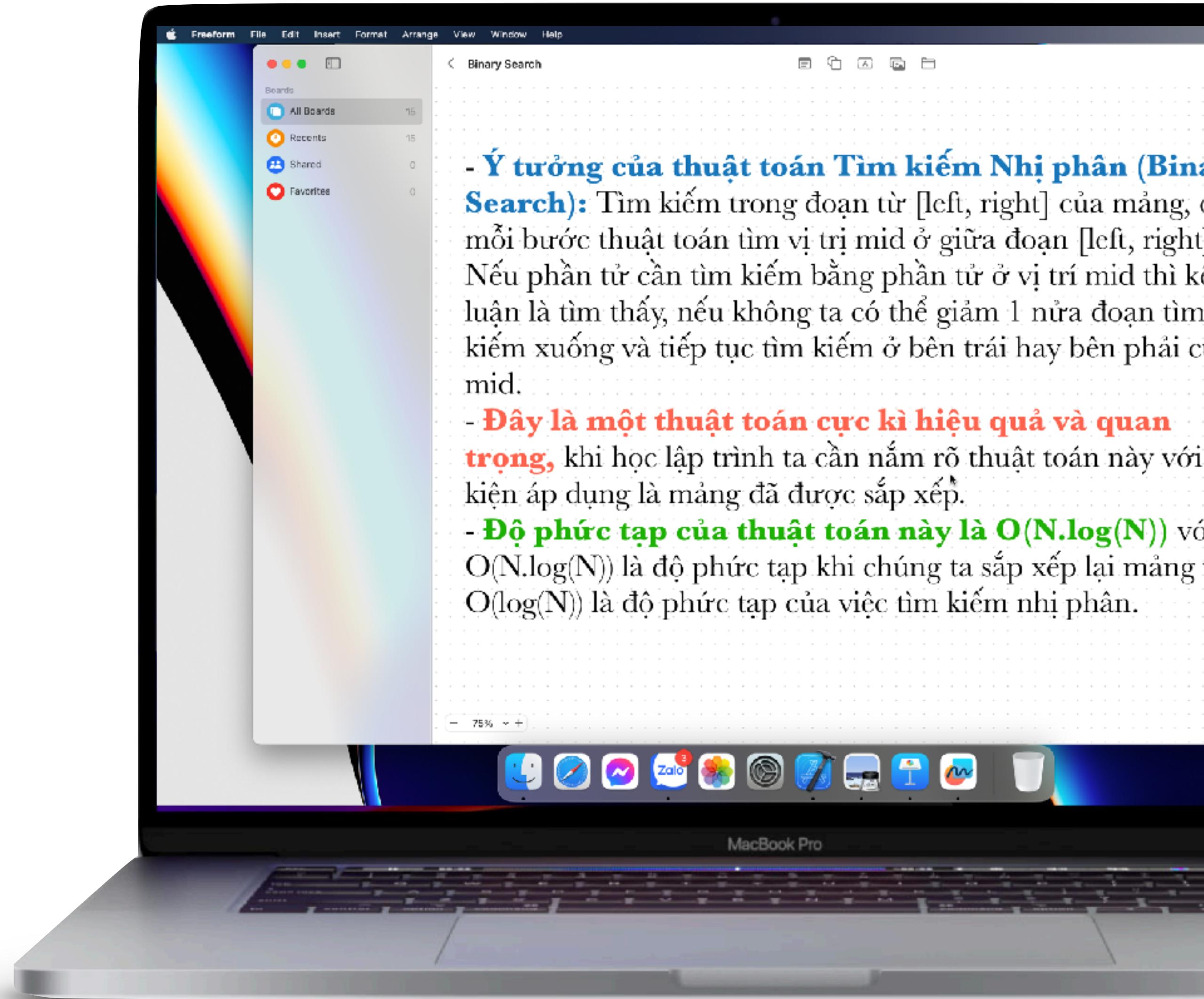
Đề 10: Câu 1

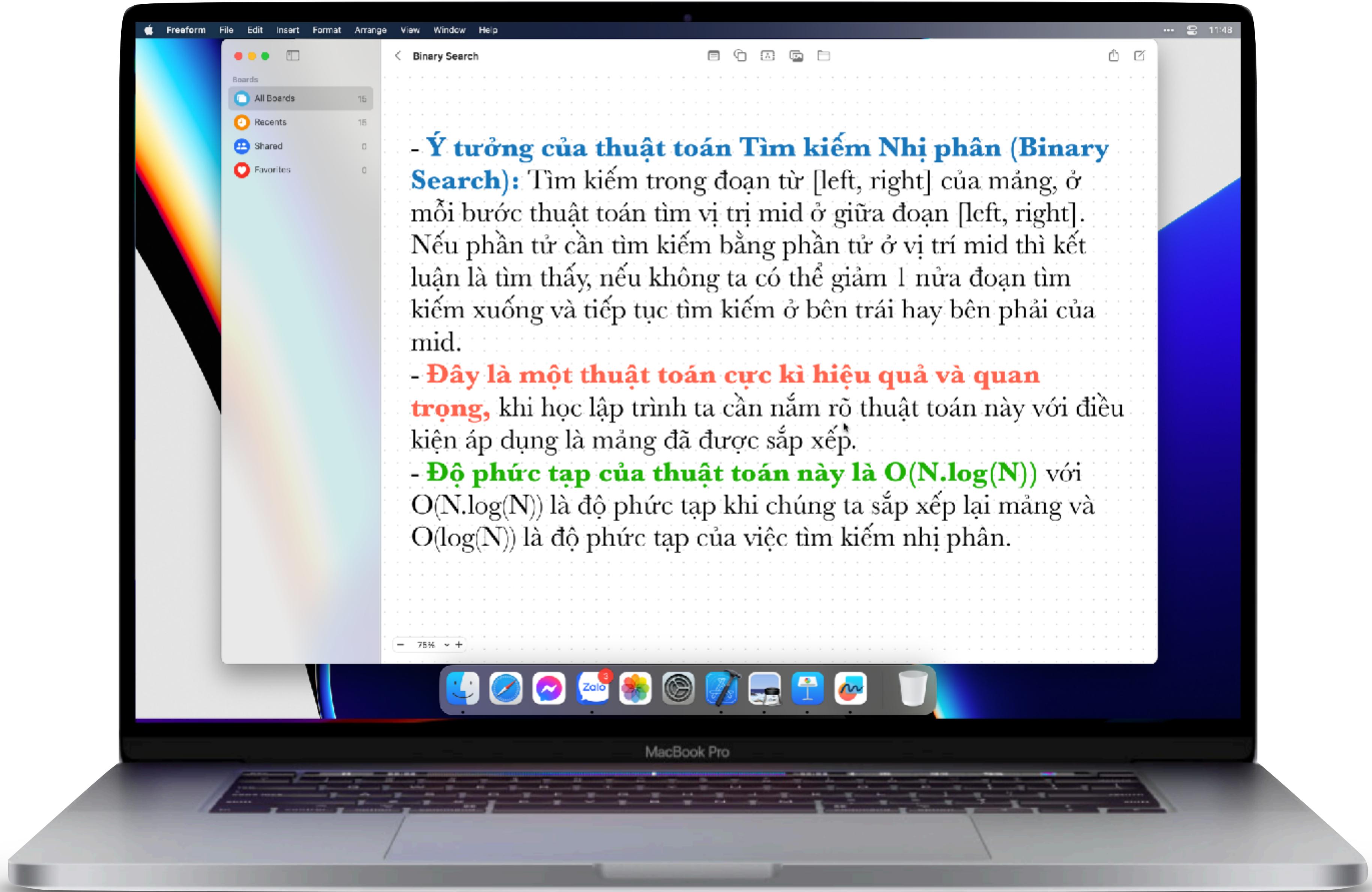
ĐỀ BÀI

Đề 10: Câu 1

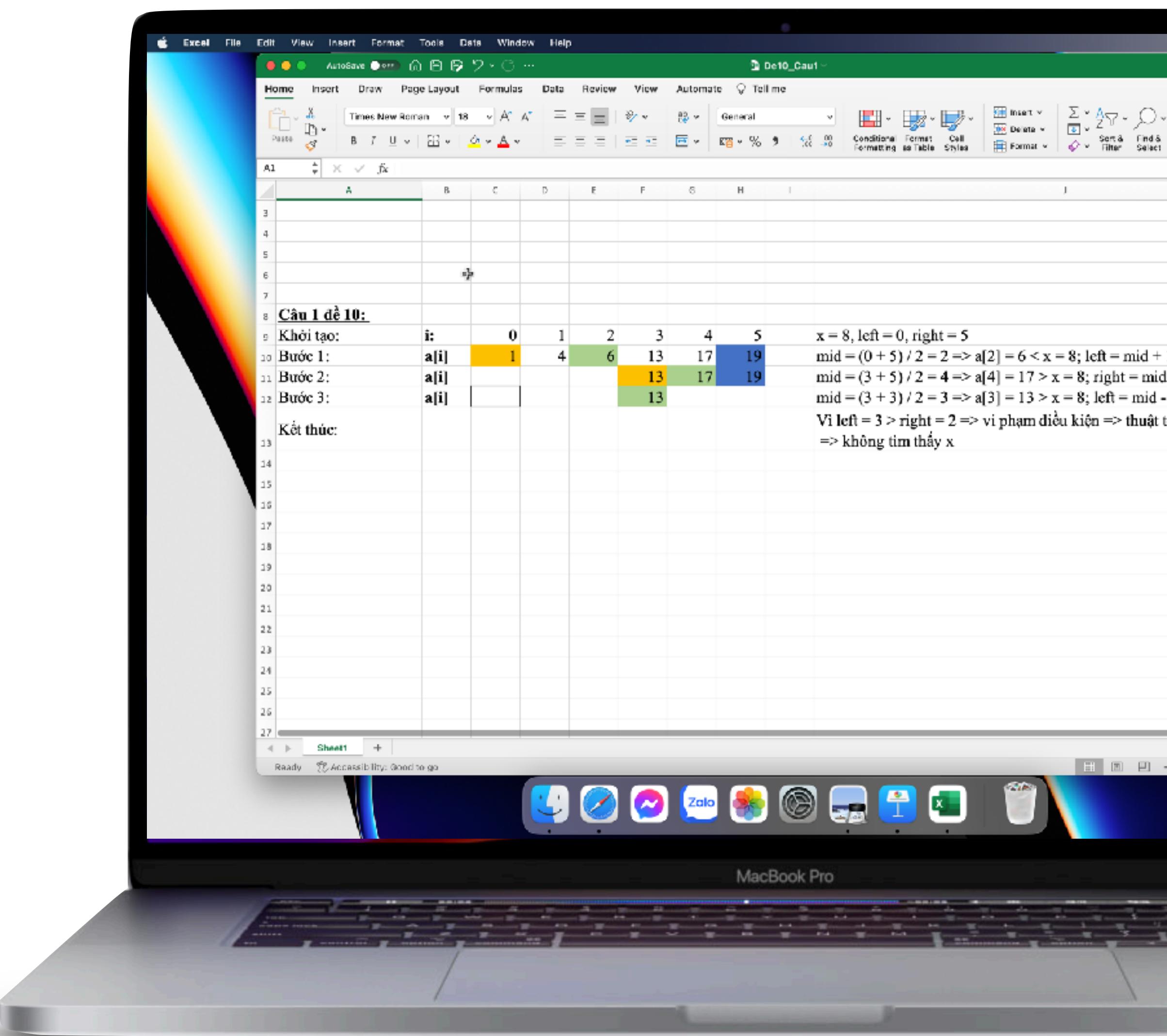
- a.** Anh/Chị hãy trình bày **ý tưởng của thuật toán** tìm kiếm nhị phân (**Binary Search**) để tìm kiếm một số trên một mảng đã có thứ tự.
- b.** Trình bày các bước (**vẽ từng bước**) theo thuật toán tìm kiếm nhị phân thực hiện tìm kiếm giá trị $x = 8$ trên mảng số nguyên có giá trị: **1, 4, 6, 13, 17, 19.**

a. Anh/Chị hãy trình bày ý tưởng của thuật toán tìm kiếm nhị phân (**Binary Search**) để tìm kiếm một số trên một mảng đã có thứ tự.



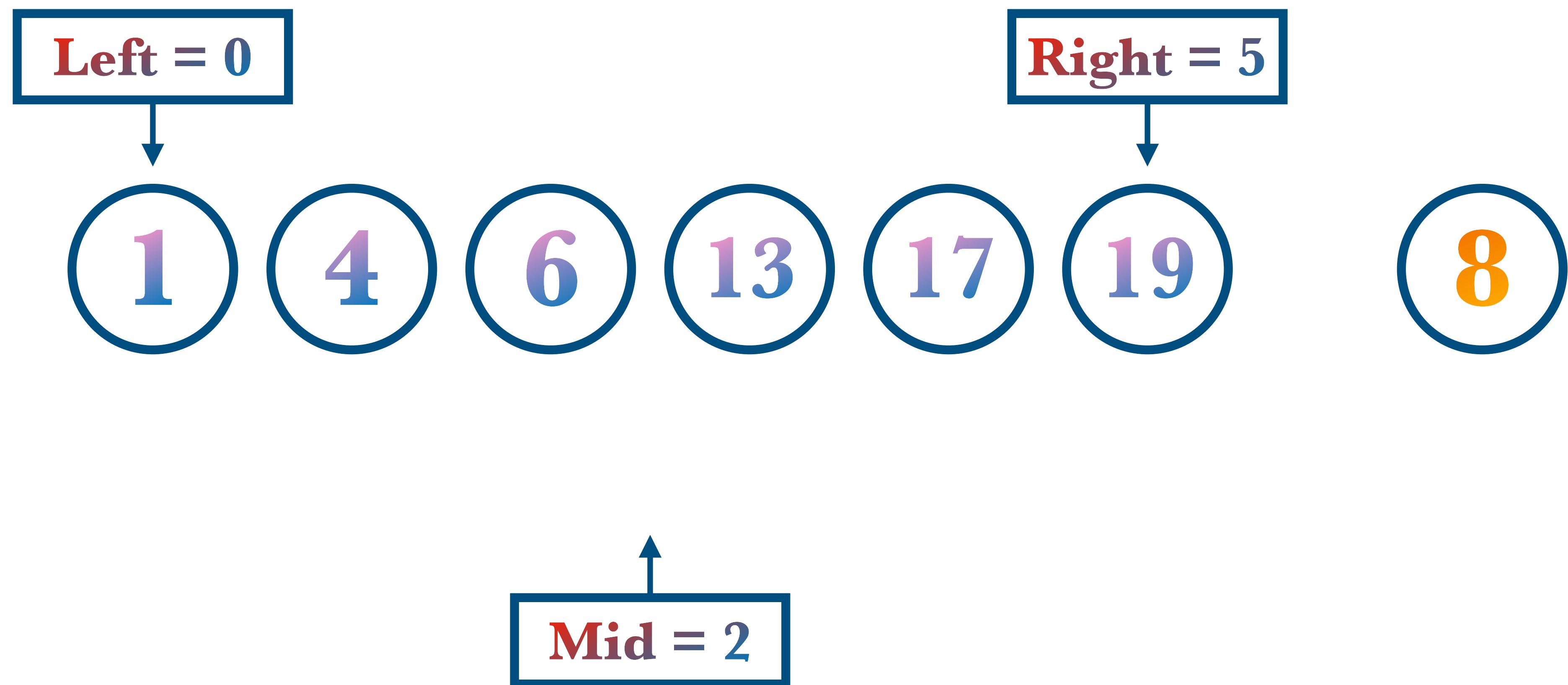


b. Trình bày các bước (**vẽ từng bước**) theo thuật toán tìm kiếm nhị phân thực hiện tìm kiếm giá trị $x = 8$ trên mảng số nguyên có giá trị: **1, 4, 6, 13, 17, 19.**



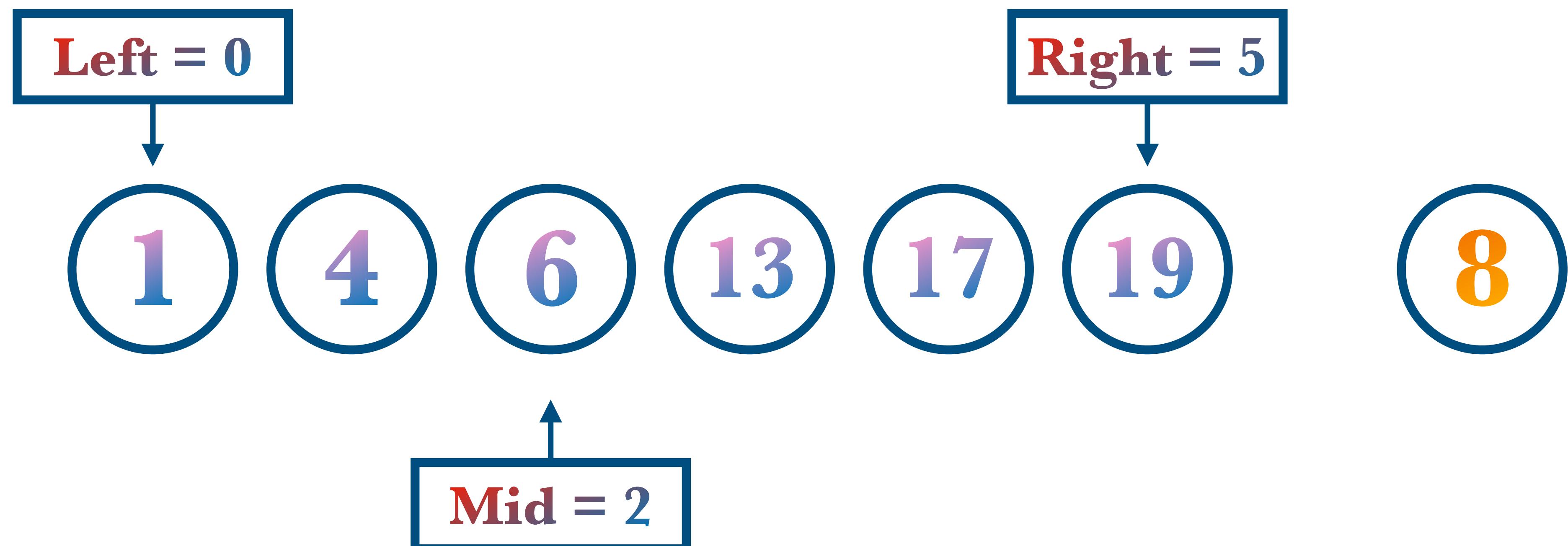
MINH HOẠ THUẬT TOÁN

BINARY SEARCH



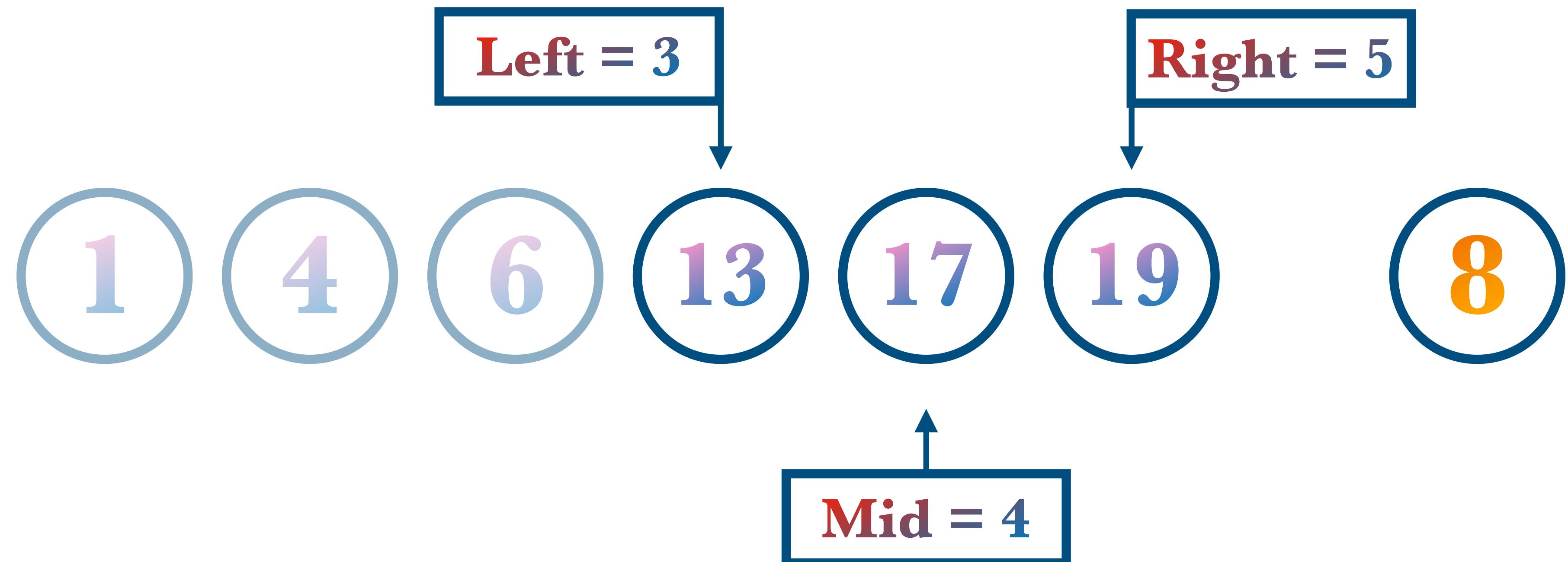
MINH HOẠ THUẬT TOÁN

BINARY SEARCH



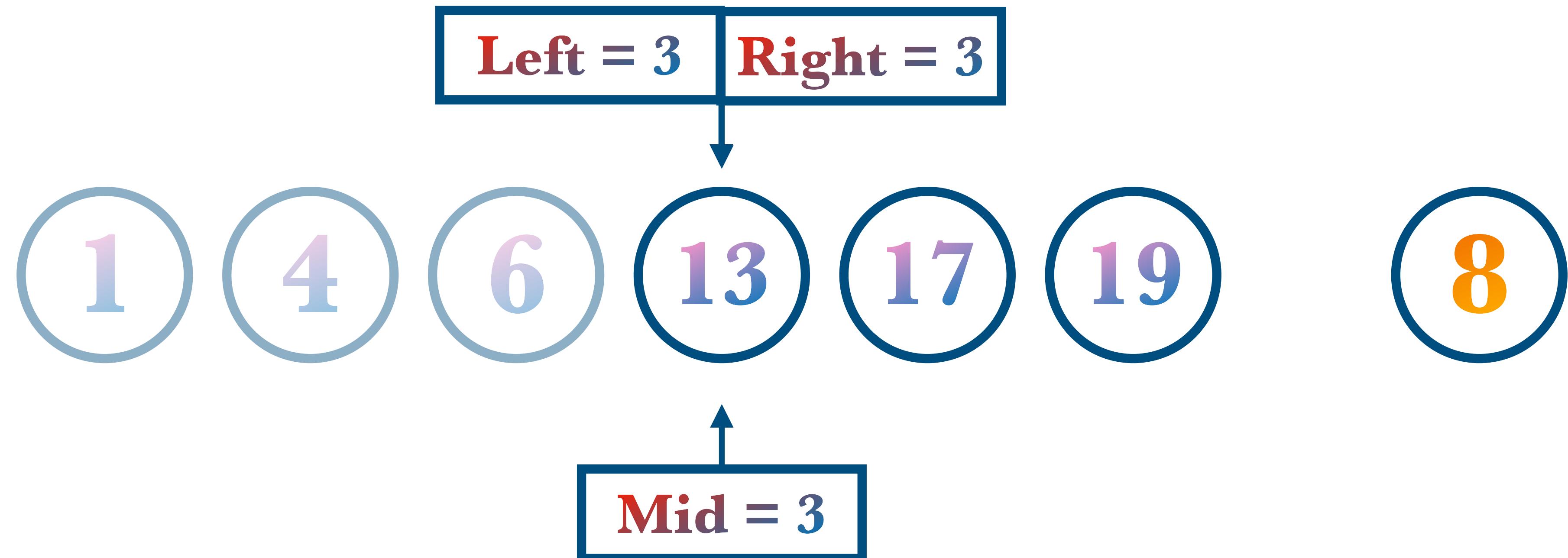
MINH HOẠ THUẬT TOÁN

BINARY SEARCH



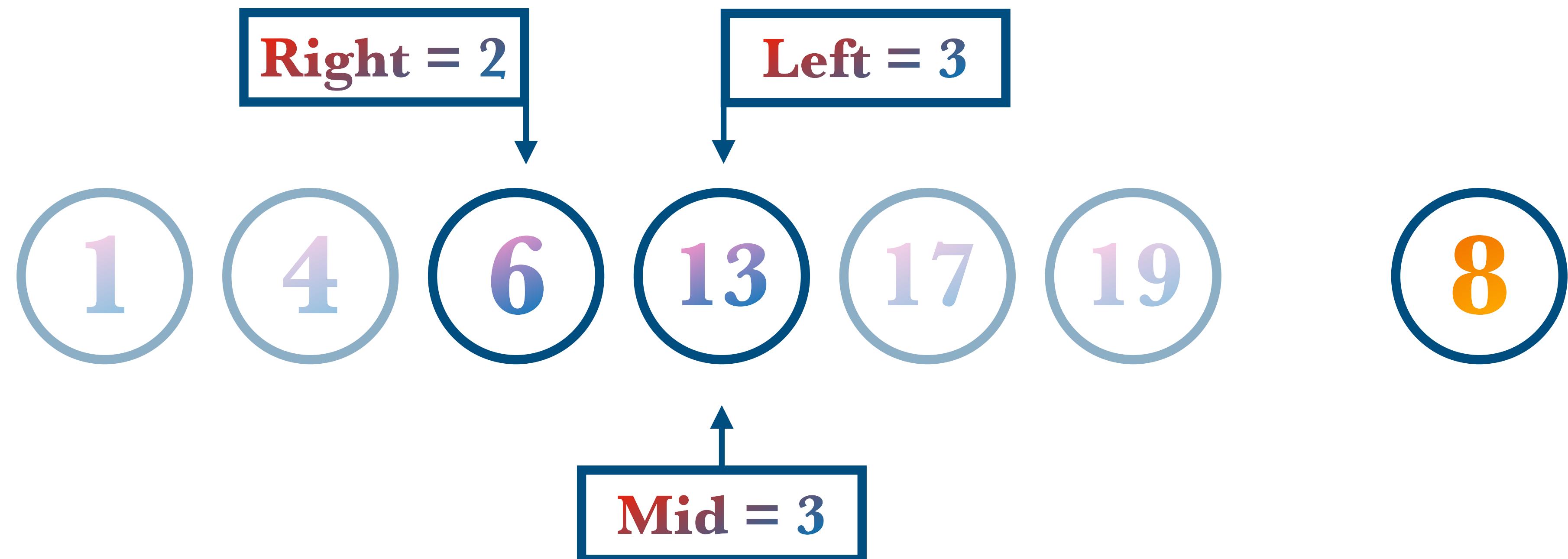
MINH HOẠ THUẬT TOÁN

BINARY SEARCH



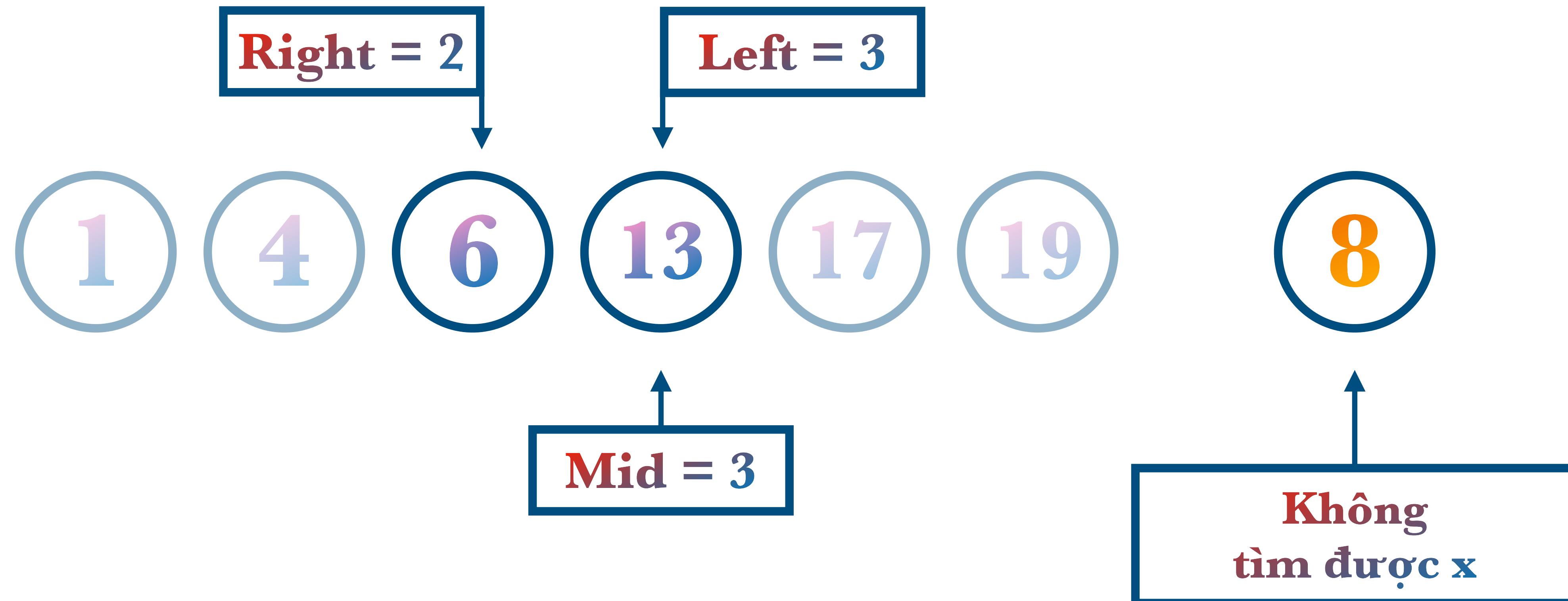
MINH HOẠ THUẬT TOÁN

BINARY SEARCH



MINH HOẠ THUẬT TOÁN

BINARY SEARCH



CÂY NHỊ PHÂN TÌM KIẾM

Đề 11: Câu 2

ĐỀ BÀI

Đề 11: Câu 2

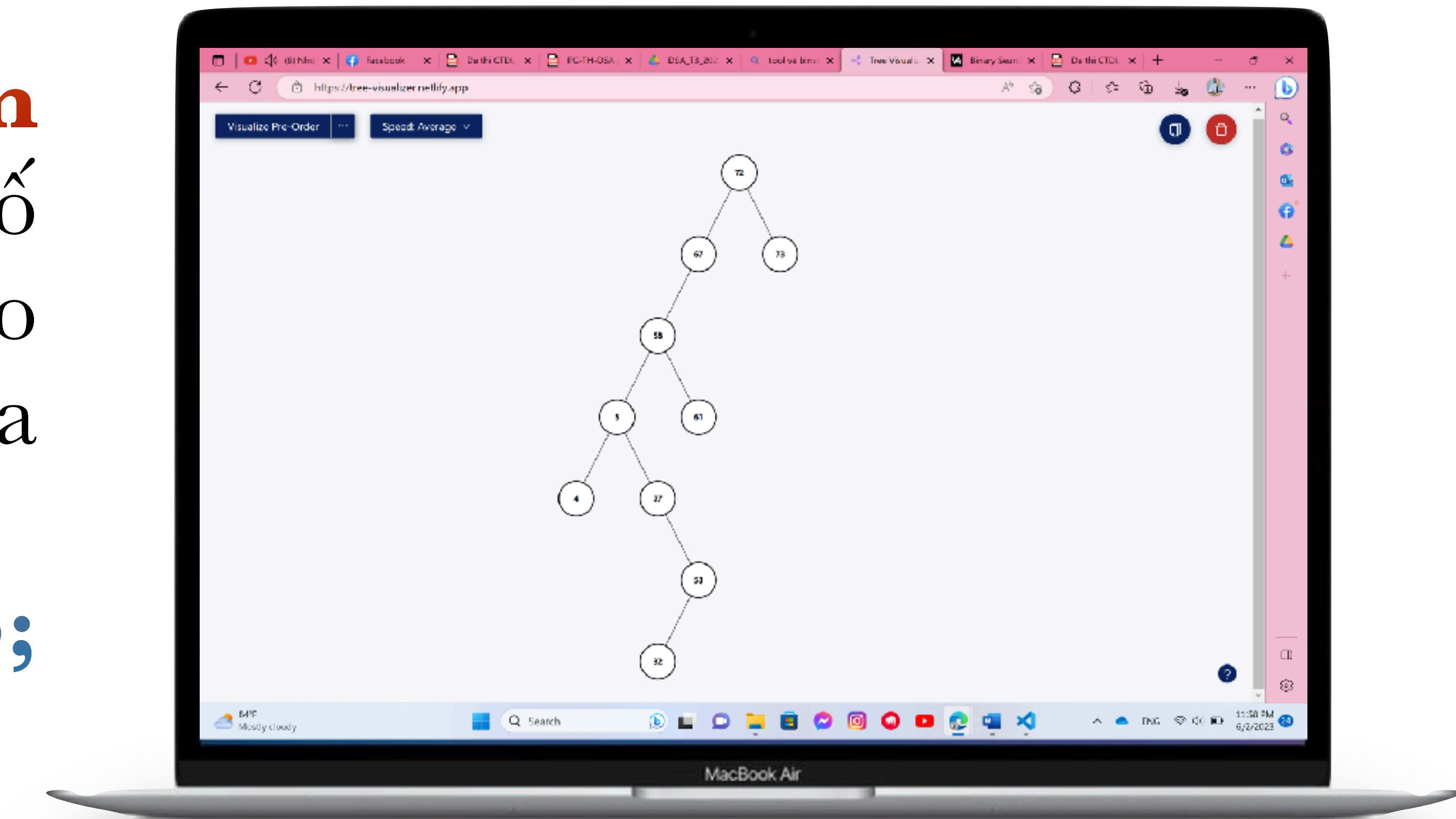
Hãy phát biểu định nghĩa **cây nhị phân tìm kiếm**.

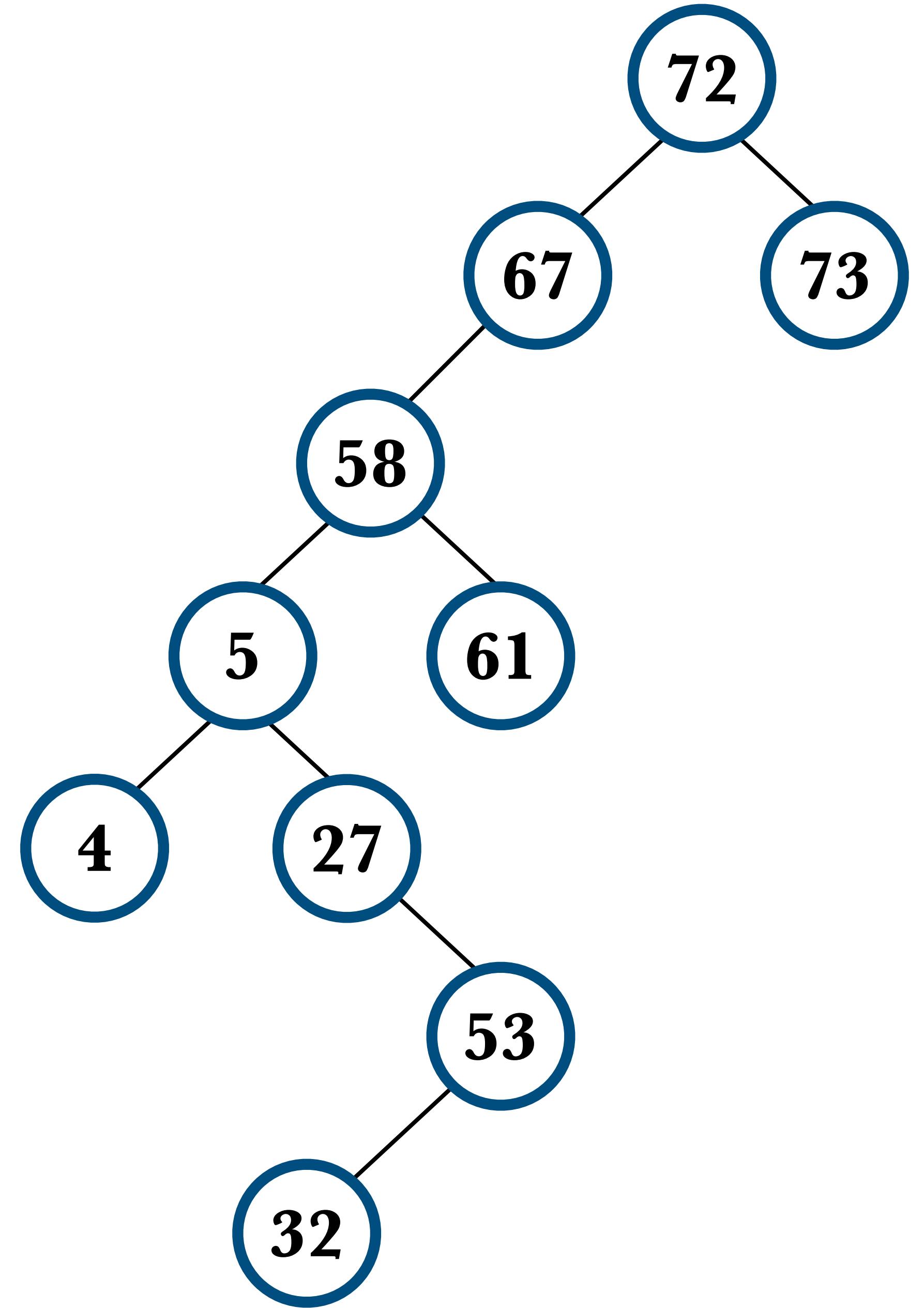
=> Cây Nhị phân Tìm kiếm (**Binary Search Tree**) là một cây nhị phân có thể tìm kiếm sự xuất hiện của 1 phần tử trên cây với **độ phức tạp $O(\log(N))$** .

Vẽ cây nhị phân tìm kiếm

(chỉ vẽ cây kết quả) từ dãy số nguyên khi xây dựng cây theo thứ tự từ trái qua phải của dãy số:

**72; 67; 73; 58; 5; 4; 27; 53;
61; 32.**





Vẽ cây nhị phân tìm kiếm
(chỉ vẽ cây kết quả) từ dãy số
nguyên khi xây dựng cây theo
thứ tự từ trái qua phải của
dãy số:
**72; 67; 73; 58; 5; 4; 27; 53;
61; 32.**

PHẦN II

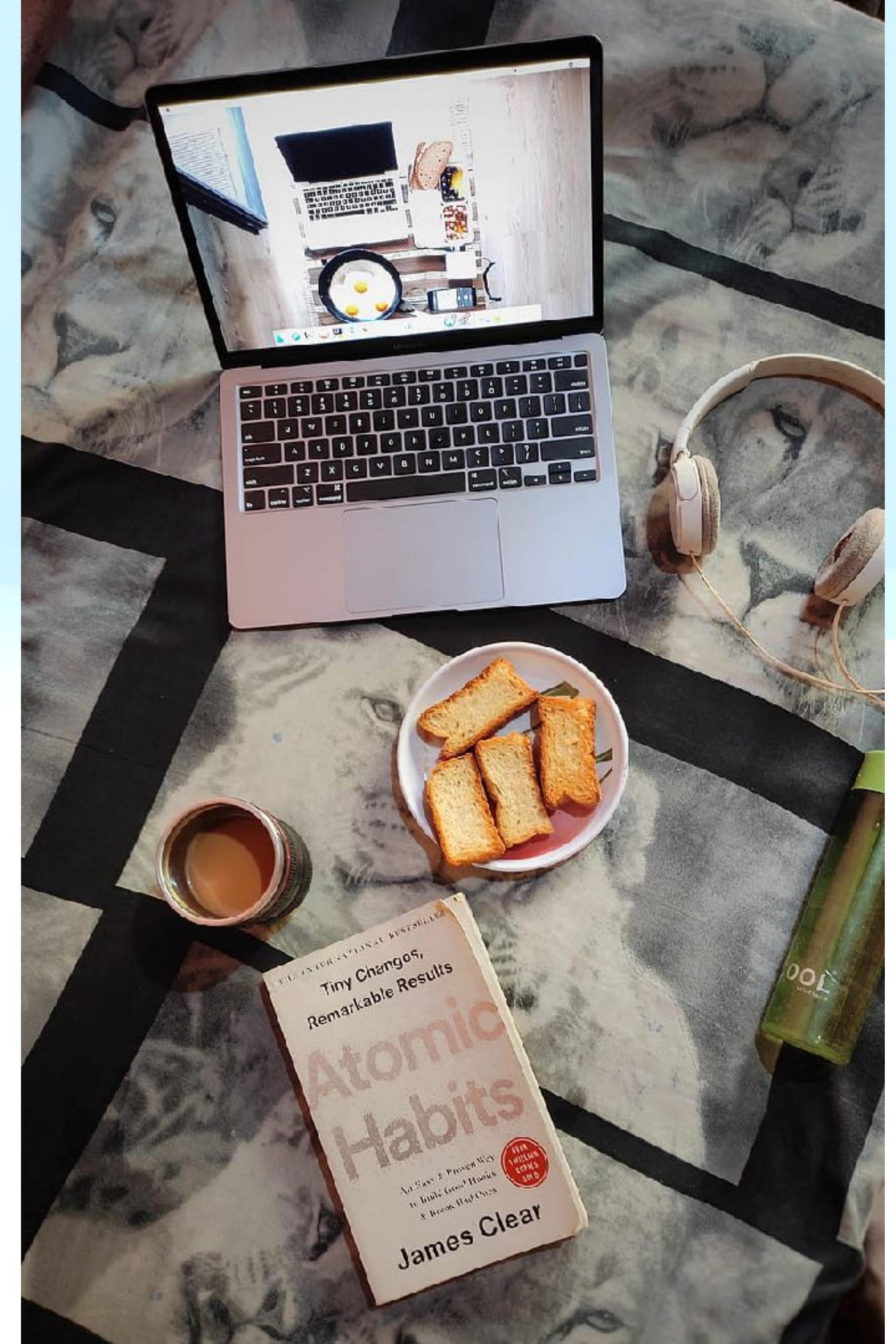
DANH SÁCH LIÊN KẾT BẢNG BĂM

Đề 10: Câu 4 - Đề 11: câu 1, 4

XEM THÊM

LẠI QUAN THIÊN

22521385



DANH SÁCH LIÊN KẾT

Đề 11: Câu 1

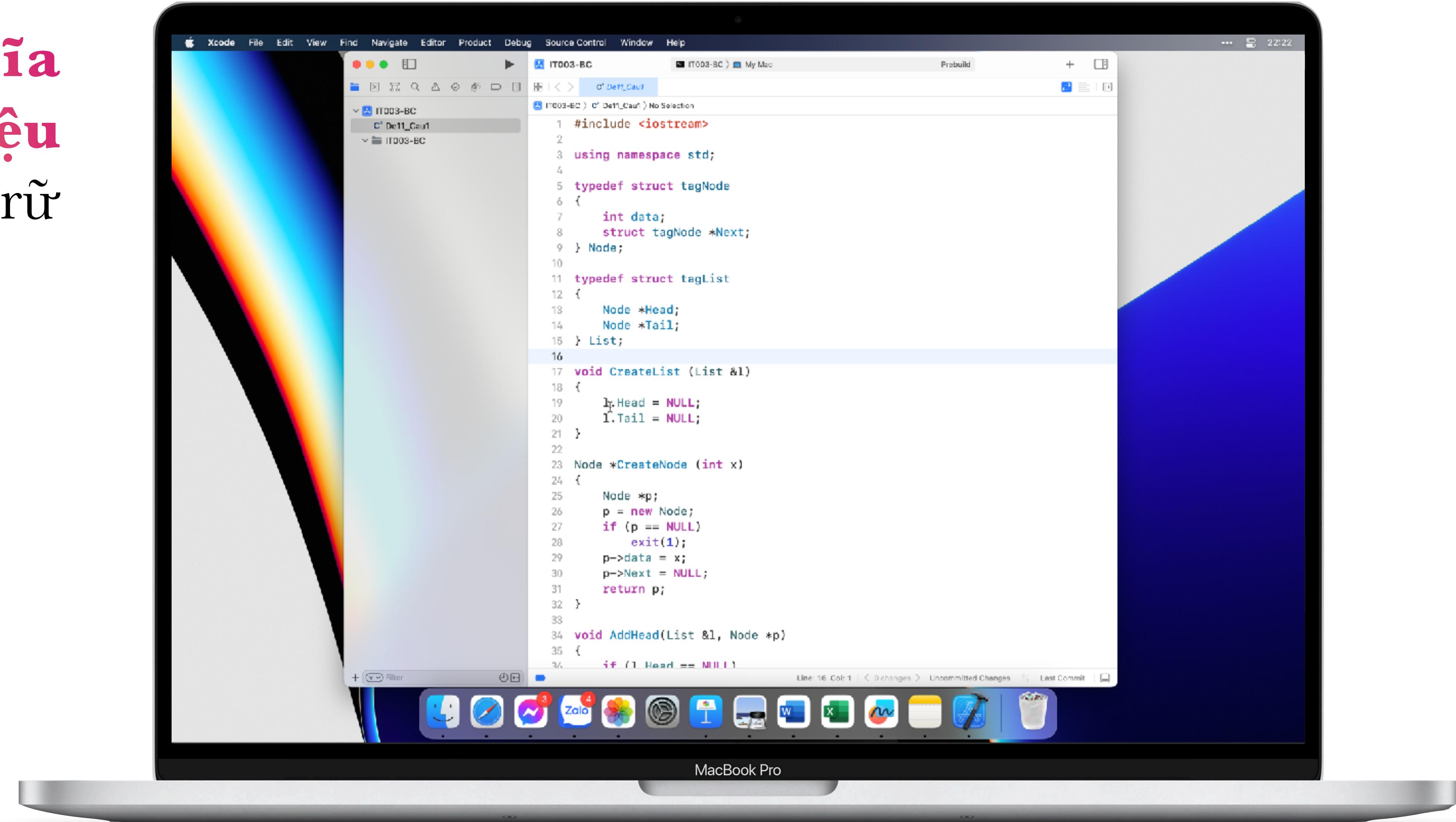
ĐỀ BÀI

Đề 11: Câu 1

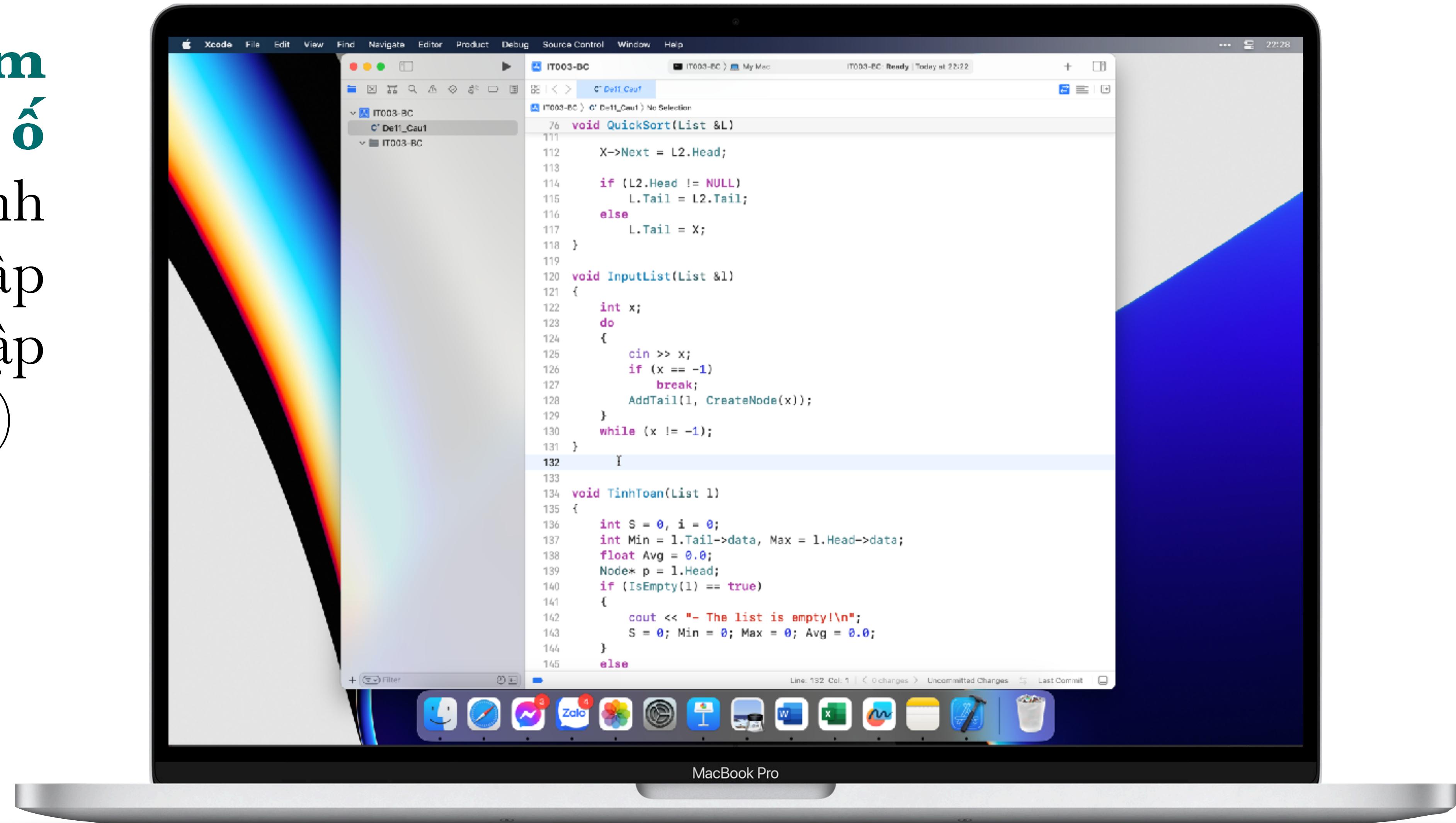
Giả sử người ta có nhu cầu dùng danh sách liên kết để **lưu trữ các số nguyên dương**. Anh/chị hãy thực hiện :

- a.** **Định nghĩa cấu trúc dữ liệu** để có thể lưu trữ như yêu cầu (1 điểm)
- b.** **Viết hàm nhập các số nguyên** vào danh sách, việc nhập kết thúc khi nhập giá trị -1 (1 điểm)
- c.** **Viết hàm void TinhToan(ptr Head)** in ra giá trị lớn nhất, giá trị nhỏ nhất và giá trị trung bình của danh sách. Qui ước: nếu danh sách là rỗng thì các giá trị này đều là 0. (1 điểm)
- d.** **Viết hàm sắp xếp danh sách** theo giá trị của phần tử trong danh sách **giảm dần** và in ra màn hình (1 điểm)

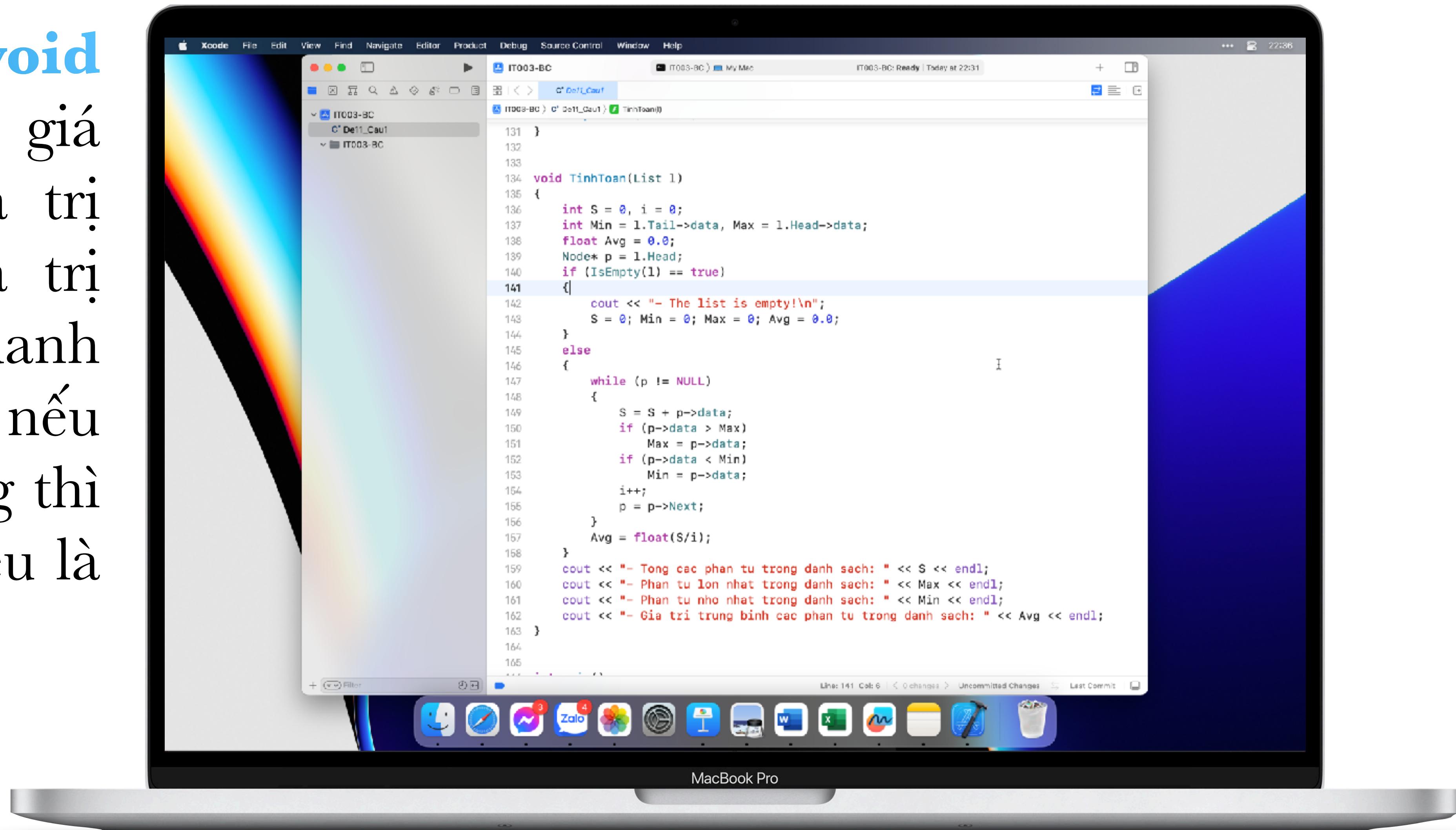
**a. Định nghĩa
cấu trúc dữ liệu
để có thể lưu trữ
như yêu cầu
(1 điểm)**



b. Viết hàm
nhập các số
nguyên vào danh
sách, việc nhập
kết thúc khi nhập
giá trị -1 (1 điểm)



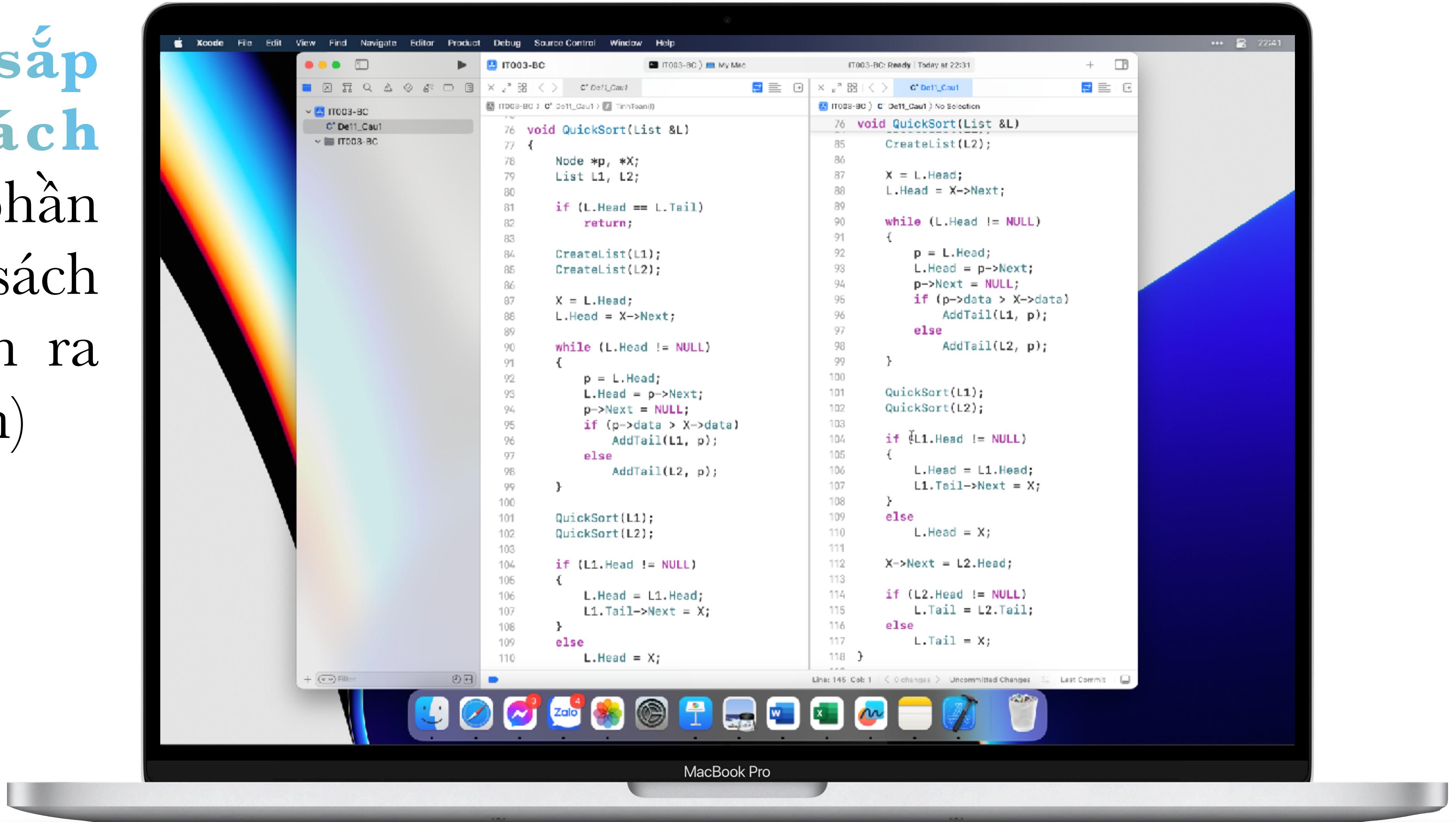
c. Viết hàm void TinhToan in ra giá trị lớn nhất, giá trị nhỏ nhất và giá trị trung bình của danh sách. Qui ước: nếu danh sách là rỗng thì các giá trị này đều là 0. (1 điểm)



The image shows a MacBook Pro with its screen displaying the Xcode IDE on a Mac OS desktop. The code in the editor is a C++ function named `TinhToan` that takes a `List` parameter. The function calculates the sum of all elements in the list, finds the maximum and minimum values, and computes the average. It handles the case where the list is empty by returning 0 for all statistical values.

```
131 }
132
133
134 void TinhToan(List l)
135 {
136     int S = 0, i = 0;
137     int Min = l.Tail->data, Max = l.Head->data;
138     float Avg = 0.0;
139     Node* p = l.Head;
140     if (IsEmpty(l) == true)
141     {}
142     cout << "- The list is empty!\n";
143     S = 0; Min = 0; Max = 0; Avg = 0.0;
144 }
145 else
146 {
147     while (p != NULL)
148     {
149         S = S + p->data;
150         if (p->data > Max)
151             Max = p->data;
152         if (p->data < Min)
153             Min = p->data;
154         i++;
155         p = p->Next;
156     }
157     Avg = float(S/i);
158 }
159 cout << "- Tong cac phan tu trong danh sach: " << S << endl;
160 cout << "- Phan tu lon nhat trong danh sach: " << Max << endl;
161 cout << "- Phan tu nho nhat trong danh sach: " << Min << endl;
162 cout << "- Gia tri trung binh cac phan tu trong danh sach: " << Avg << endl;
163 }
164
165 }
```

d. Viết hàm **sắp xếp danh sách** theo giá trị của phần tử trong danh sách **giảm dần** và in ra màn hình (1 điểm)



BẢNG BĂM

Đề 10: Câu 4 - Đề 11: Câu 4

ĐỀ BÀI

Đề 10: Câu 4

a. Anh/Chị hãy cho biết **ưu điểm bảng băm** khi so với cấu trúc dữ liệu đã học (mảng, danh sách liên kết đơn, cây).

b. Cho bảng băm A kích thước **7 phần tử** và tập khóa

$$K = \{76, 40, 48, 5, 20\}.$$

Hãy vẽ bảng băm khi thêm từng khóa K vào bảng A với hàm băm

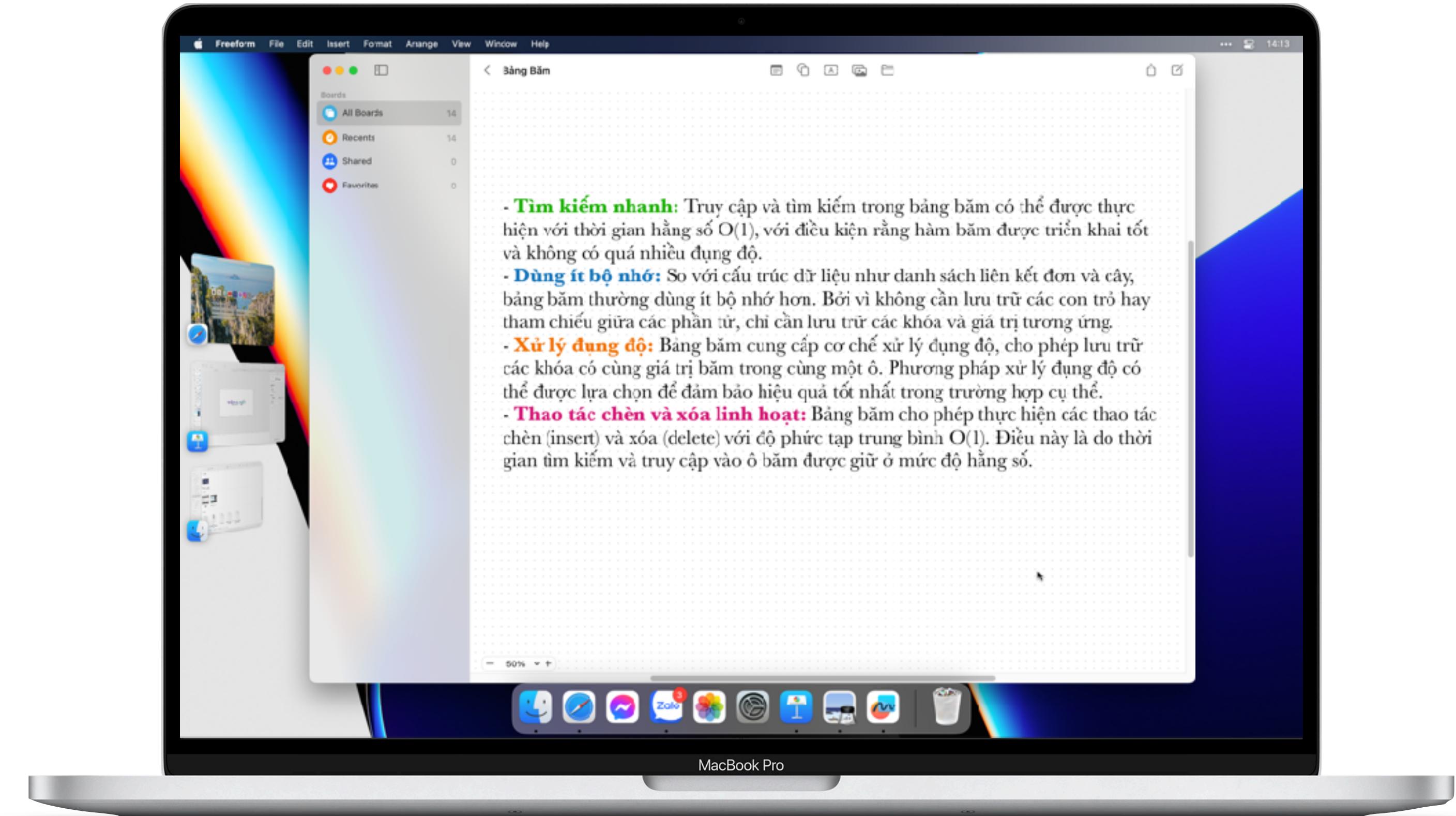
$$hf(k) = k \% 7.$$

trong trường hợp xảy ra đụng độ, hãy sử dụng **phương pháp dò bậc hai (Quadratic Probing Method)** để giải quyết đụng độ.

ĐỀ BÀI

Đề 10: Câu 4

a. Anh/Chị hãy cho biết **ưu điểm bảng băm** khi so với cấu trúc dữ liệu đã học (mảng, danh sách liên kết đơn, cây).



b. Cho bảng băm A kích thước **7 phần tử** và tập khóa

$$K = \{76, 40, 48, 5, 20\}.$$

Hãy vẽ bảng băm khi thêm từng khóa K vào bảng A với hàm băm

$$hf(k) = k \% 7.$$

trong trường hợp xảy ra đụng độ, hãy sử dụng **phương pháp dò bậc hai (Quadratic Probing Method)** để giải quyết đụng độ.

b. Cho bảng băm A kích thước **7 phần tử** và tập khóa

Hãy vẽ bảng băm khi thêm từng khóa K vào bảng A với hàm băm

trong trường hợp **K = {76, 40, 48, 5, 20}** pháp dò bậc
hai (Quadratic Probing Method) để giải quyết đụng độ.

$$hf(k) = k \% 7$$

$$hf_i(k) = [hf(k) + i^2] \% 7$$

MINH HOA

$K = \{76, 40, 48, 5, 20\}$

$$hf(k) = k \% 7$$

$$hfi(k) = [hf(k) + i^2] \% 7$$

0

1

2

3

4

5

6

$$K = \{76, 40, 48, 5, 20\}.$$

$$hf(k) = k \% 7$$

$$hfi(k) = [hf(k) + i^2] \% 7$$

0		48	48	48	48
1					
2			5	5	5
3				20	20
4					
5	40	40	40	40	40
6	76	76	76	76	76

MINH HOA

$$K = \{76, 40, 48, 5, 20\}$$

$$hf(k) = k \% 7$$

$$hfi(k) = [hf(k) + i^2] \% 7$$

0

1

2

3

4

5

6

48

5

20

48

76

IT003-BC

IT003-BC > My Mac

IT003-BC: Ready | Today at 00:36

C⁺ De11_Cau4

IT003-BC > C⁺ De11_Cau4 > No Selection

22 }

23

24 // Hàm thêm khóa vào bảng băm

25 void insertKey(int k)

26 {

27 int index = hashFunction(k); // Băm khóa vào vị trí ban đầu

28

29 // Kiểm tra xem vị trí ban đầu có trống hay không

30 if (HashTable[index].key == 0)

31 HashTable[index].key = k;

32

33 else

34 {

35 // Bước nhảy để xử lý đụng độ

36 int i = 1;

37 while (i < M)

38 {

39 int index2 = (index + i*i) % M; // Tính toán vị trí mới

40

41 // Kiểm tra xem vị trí mới có trống hay không

42 if (HashTable[index2].key == 0)

43 {

44 HashTable[index2].key = k;

45 break;

46 }

47 i++;
 }

48 }

49

50 if (i == M)
 {
 cout << "Khong the them khoa " << k << " vao bang bam." << endl;
 }
 }

54 }

55 }

56 }

ĐỀ BÀI

Đề 11: Câu 4

Giả sử cho **bảng băm A** kích thước 7 ô và **tập khóa**:

$$K = \{76, 93, 40, 47, 10, 55\}$$

Ta cần nạp các giá trị khóa K vào bảng A sử dụng hàm băm:

$$H1(k) = k \% 7.$$

Hãy vẽ bảng băm kết quả, trong trường hợp xảy ra đụng độ, hãy sử dụng **phương pháp băm kép** để xử lý, với hàm băm thứ 2 do anh/chị tự đề nghị.

ĐỀ BÀI

Đề 11: Câu 4

Giả sử cho $K = \{76, 93, 40, 47, 10, 55\}$

Ta cần nạp các giá trị $H_1(k) = k \% 7 = a$ làm băm:

Hãy vẽ bảng băm kết quả, trong trường hợp xảy ra đụng độ, hãy sử dụng **phương pháp băm kép** để xử lý, với hàm băm thứ 2 do anh/chị tự đề nghị. $H_i(k) = (a + b * i) \% 7$

MINH HOA

$$K = \{76, 93, 40, 47, 10, 55\}$$

$$H1(k) = k \% 7 = a \quad H2(k) = 5 - (k \% 5) = b \quad Hi(k) = (a + b * i) \% 7$$

0

1

2

3

4

5

6

$$K = \{76, 93, 40, 47, 10, 55\}$$

$$\begin{aligned}H1(k) &= k \% 7 = a \\H2(k) &= 5 - (k \% 5) = b \\Hi(k) &= (a + b*i) \% 7\end{aligned}$$

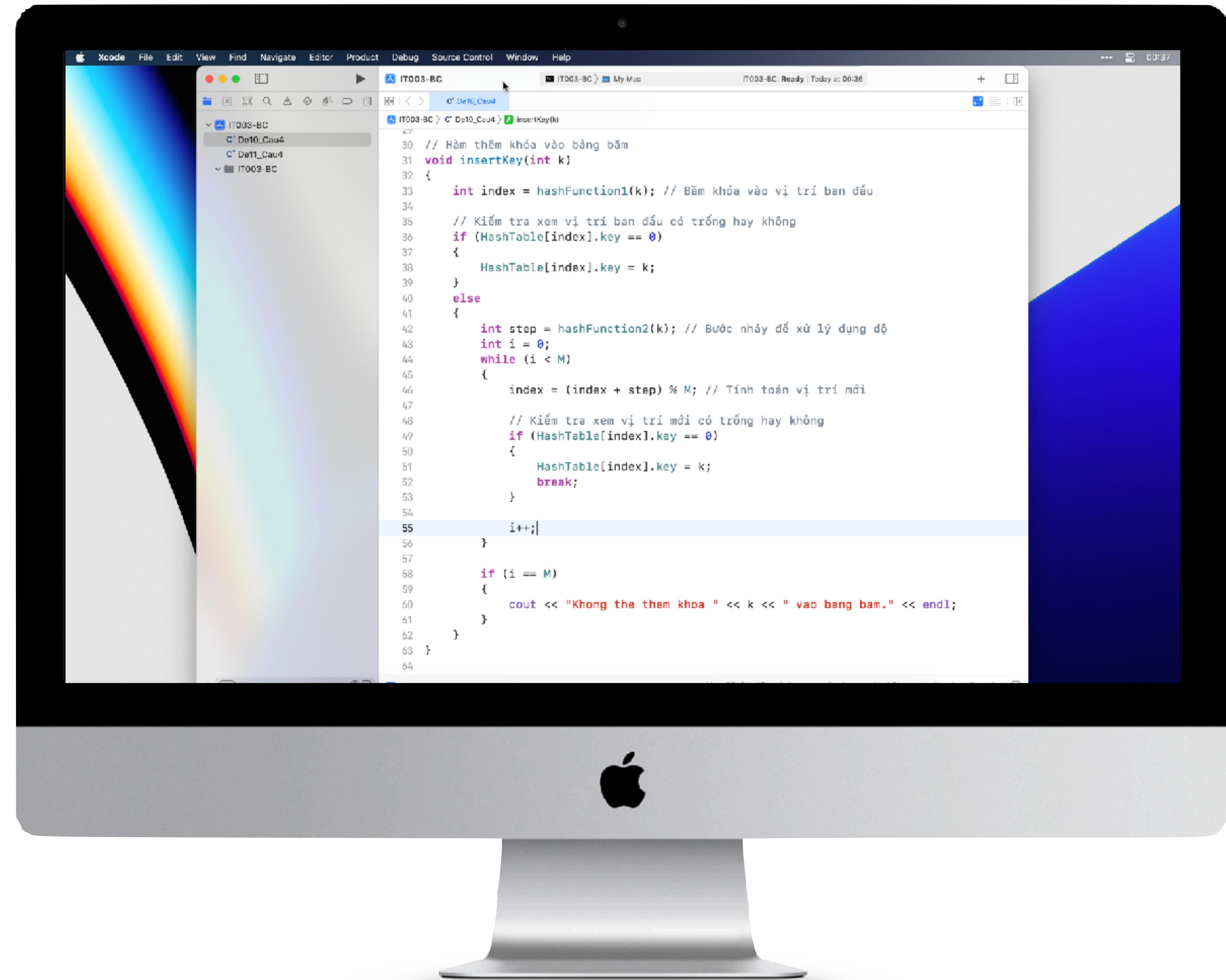
0					
1		47	47	47	47
2	93	93	93	93	93
3			10	10	10
4				55	55
5	40	40	40	40	40
6	76	76	76	76	76

MINH HOA

$$K = \{76, 93, 40, 47, 10, 55\}$$

$$H1(k) = k \% 7 = a \quad H2(k) = 5 - (k \% 5) = b \quad Hi(k) = (a + b * i) \% 7$$

0	1	2	3	4	5	6
	47	93	10	55	40	76



PHẦN III

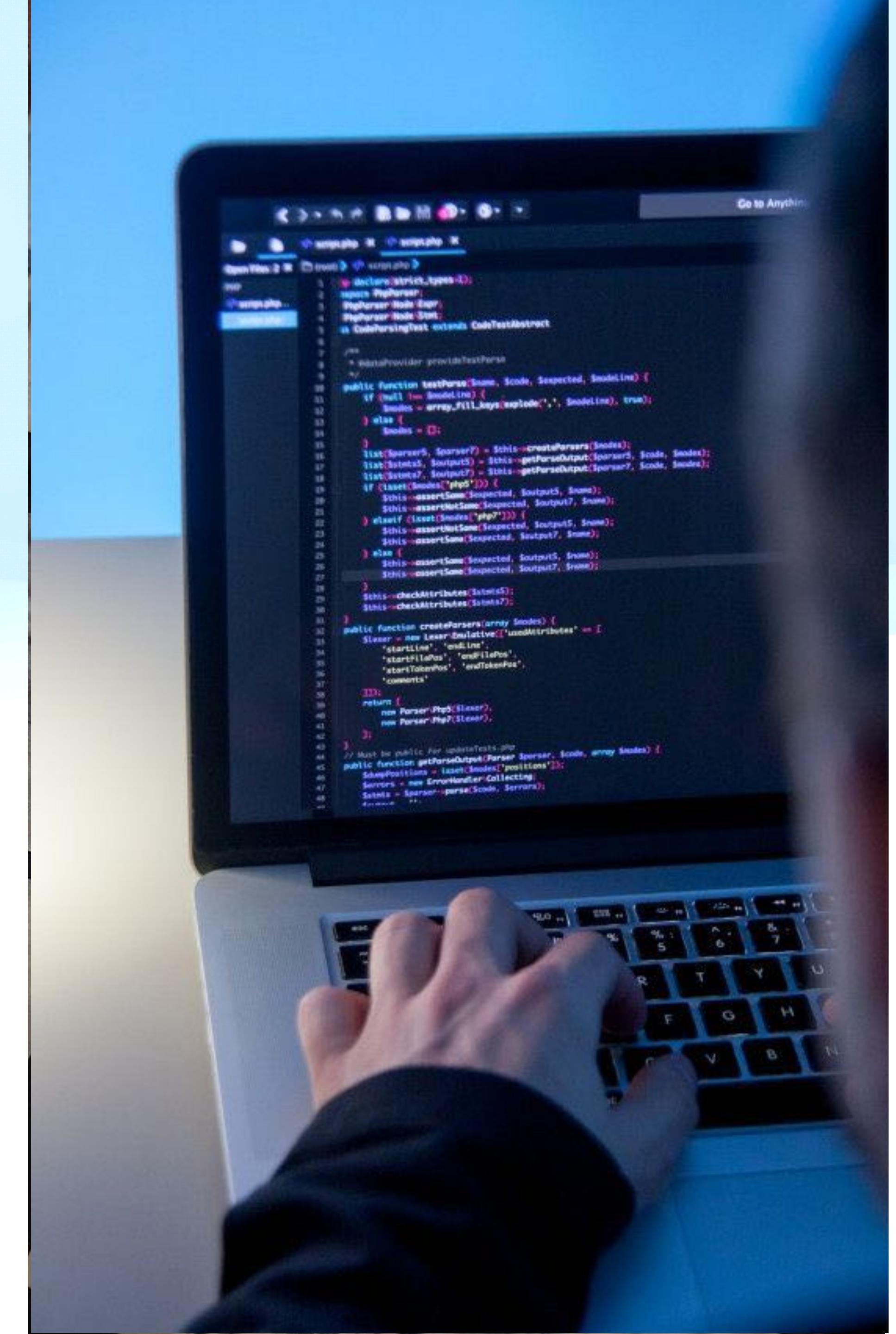
CÂY NHÌ PHÂN TÌM KIẾM

Đề 10: Câu 3 - Đề 11: câu 3

X E M T H È M

TRẦN THẾ HỮU PHÚC

22521143



CÂY NHỊ PHÂN TÌM KIẾM

Đề 10: Câu 3

ĐỀ BÀI

Đề 10: Câu 3

Biết rằng khi duyệt cây nhị phân tìm kiếm T theo thứ tự **Left - Right - Node (LRN)** thì được dãy sau:

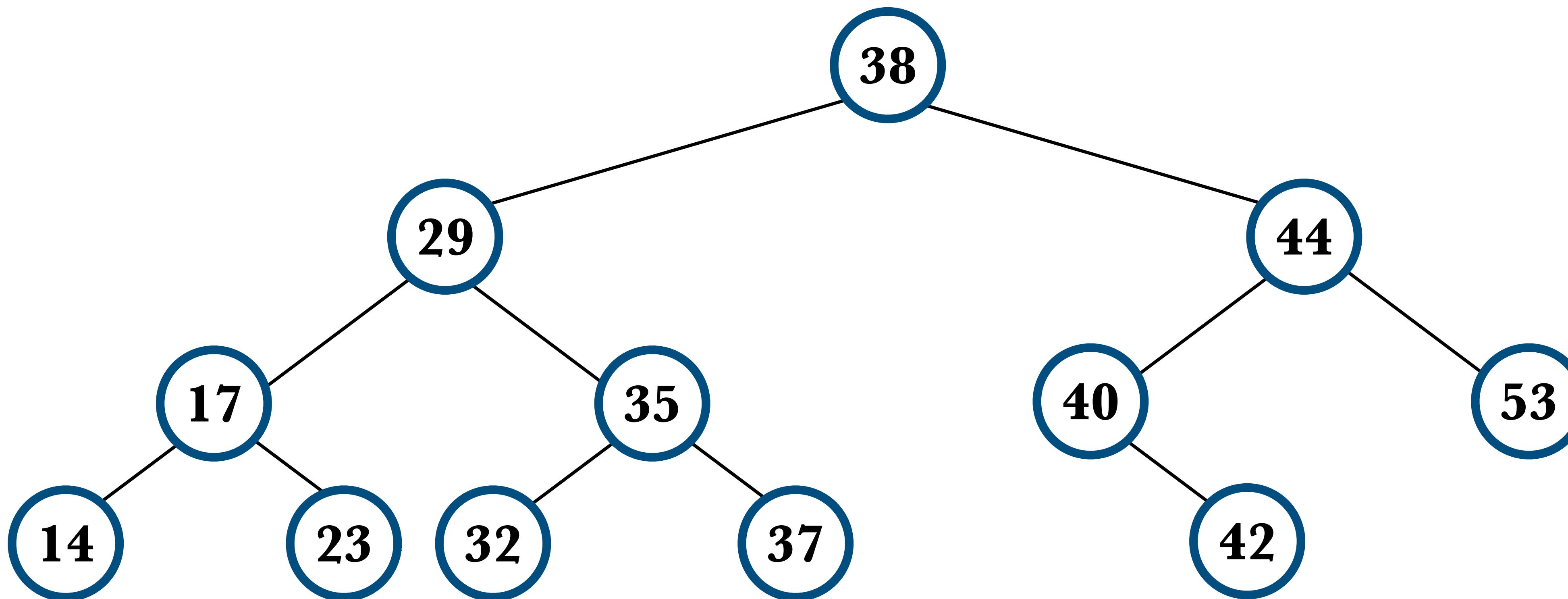
14, 23, 17, 32, 37, 35, 29, 42, 40, 53, 44, 38.

Hãy thực hiện các yêu cầu sau:

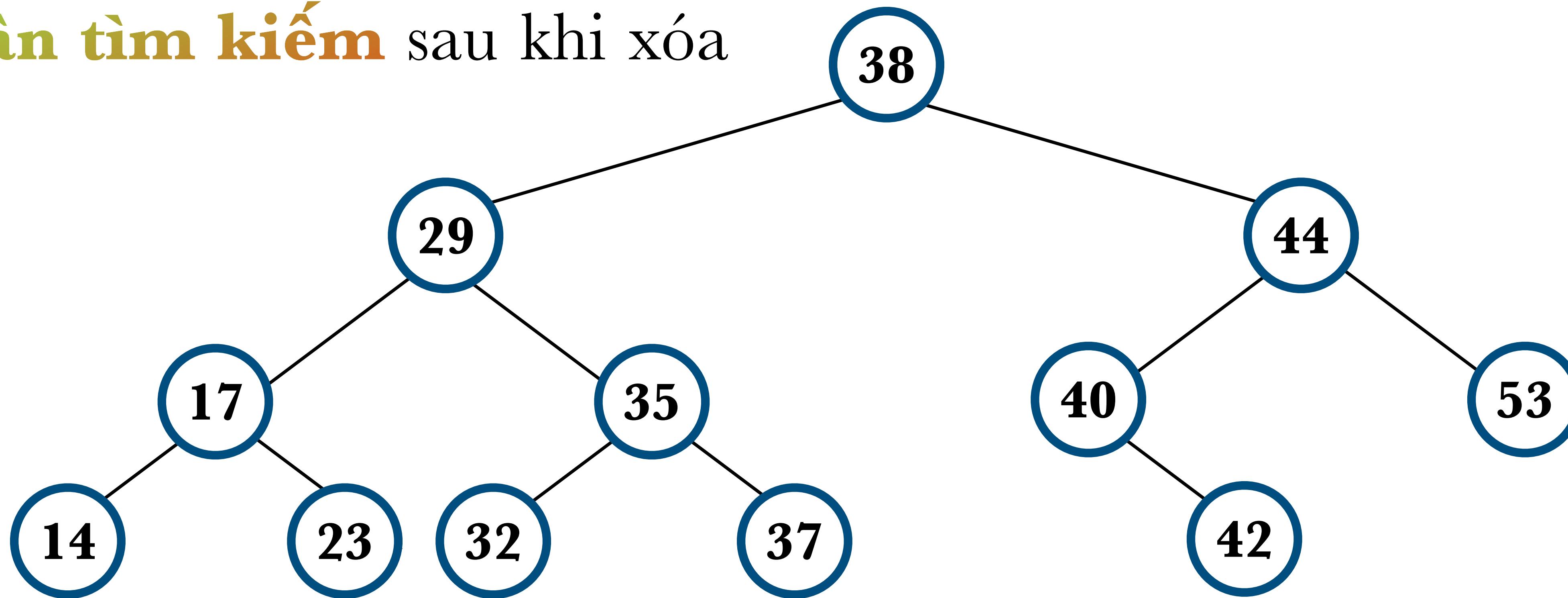
- a. Vẽ lại** cây nhị phân tìm kiếm T.
- b. Xóa khỏi cây T** lần lượt các nút **37, 40, 29, 38** (vẽ hình từng trường hợp) sao cho cây **vẫn là cây nhị phân tìm kiếm** sau khi xóa nút.
- c. Viết hàm** in ra **tổng giá trị các nút ở mức k** trên cây nhị phân tìm kiếm T (với quy ước mức ở nút gốc là 0).

a. Vẽ lại cây nhị phân tìm kiếm T.

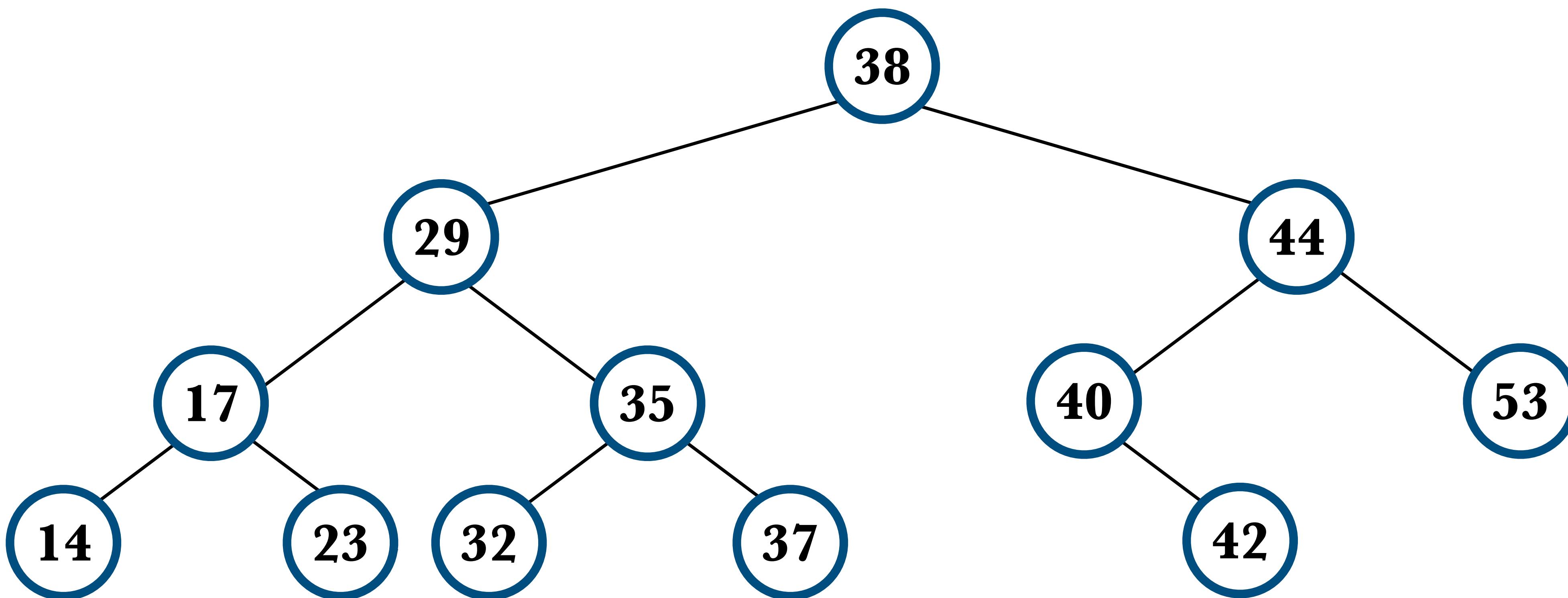
14, 23, 17, 32, 37, 35, 29, 42, 40, 53, 44, 38.



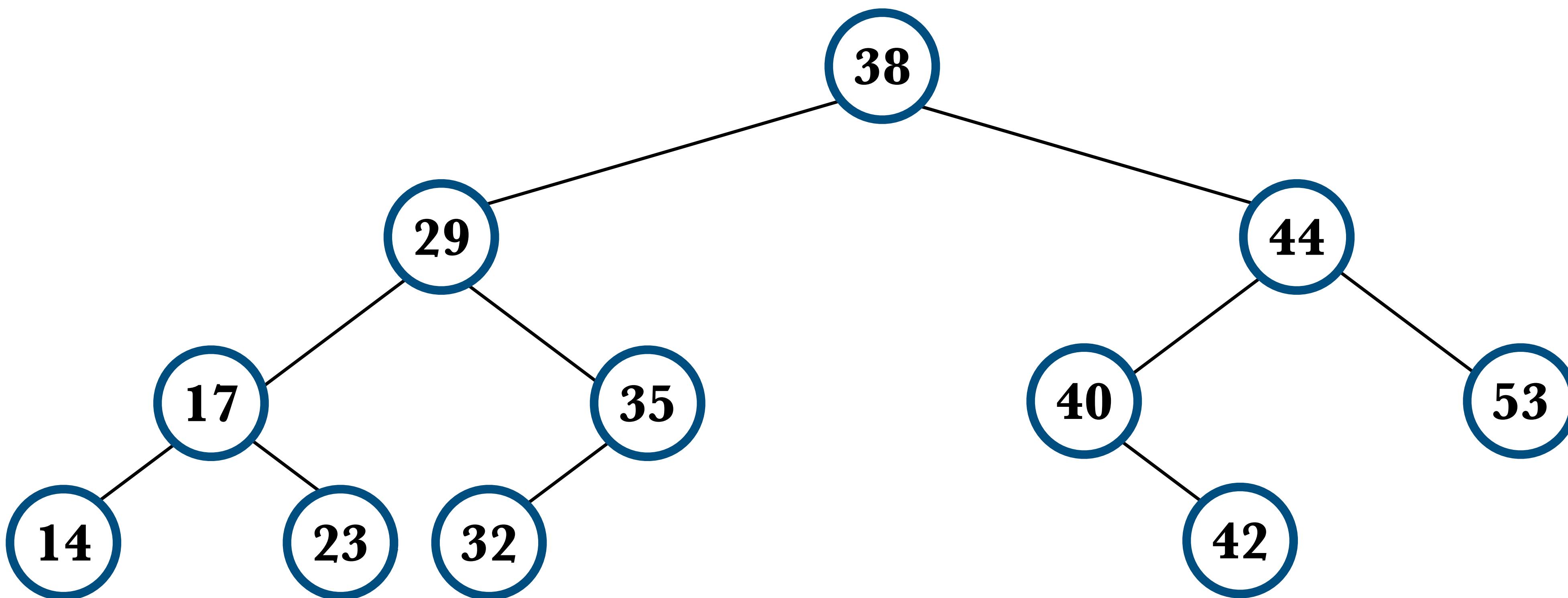
b. Xóa khỏi cây T lần lượt
các nút **37, 40, 29, 38** (vẽ
hình từng trường hợp) sao
cho cây **vẫn là cây nhị
phân tìm kiếm** sau khi xóa
nút.



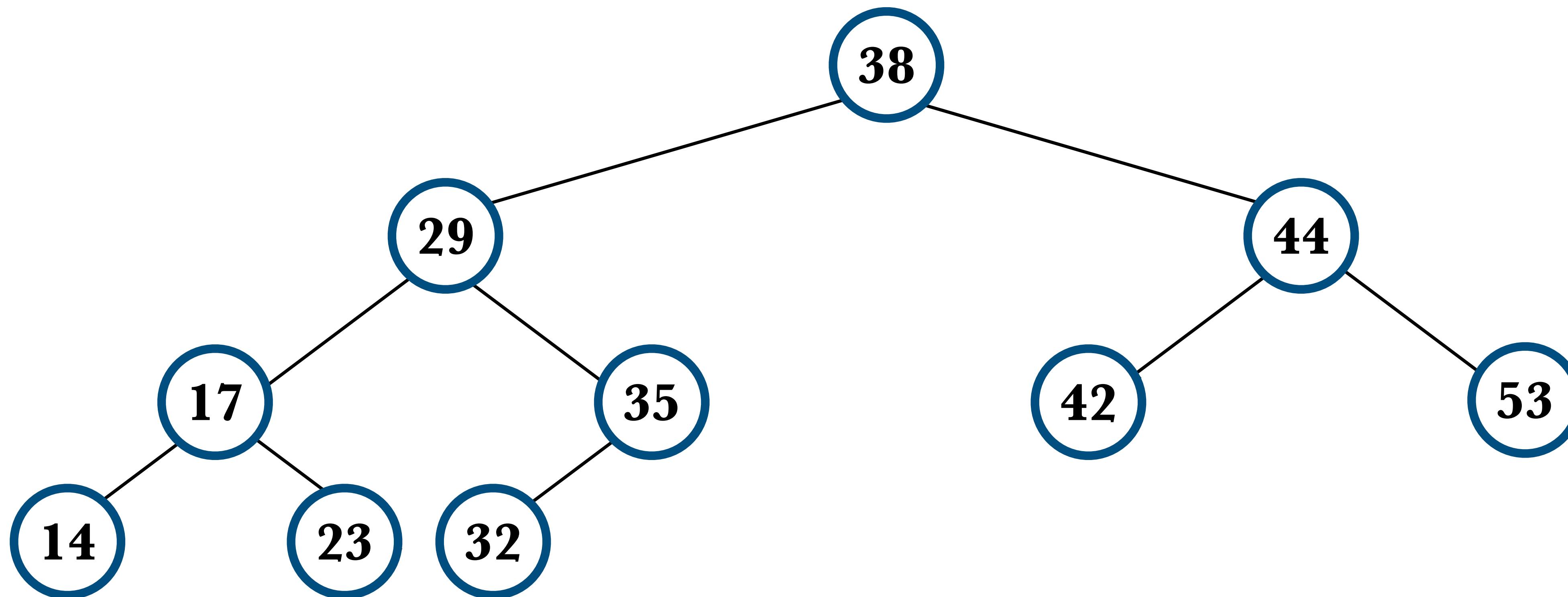
Xóa khỏi cây T nút 37:



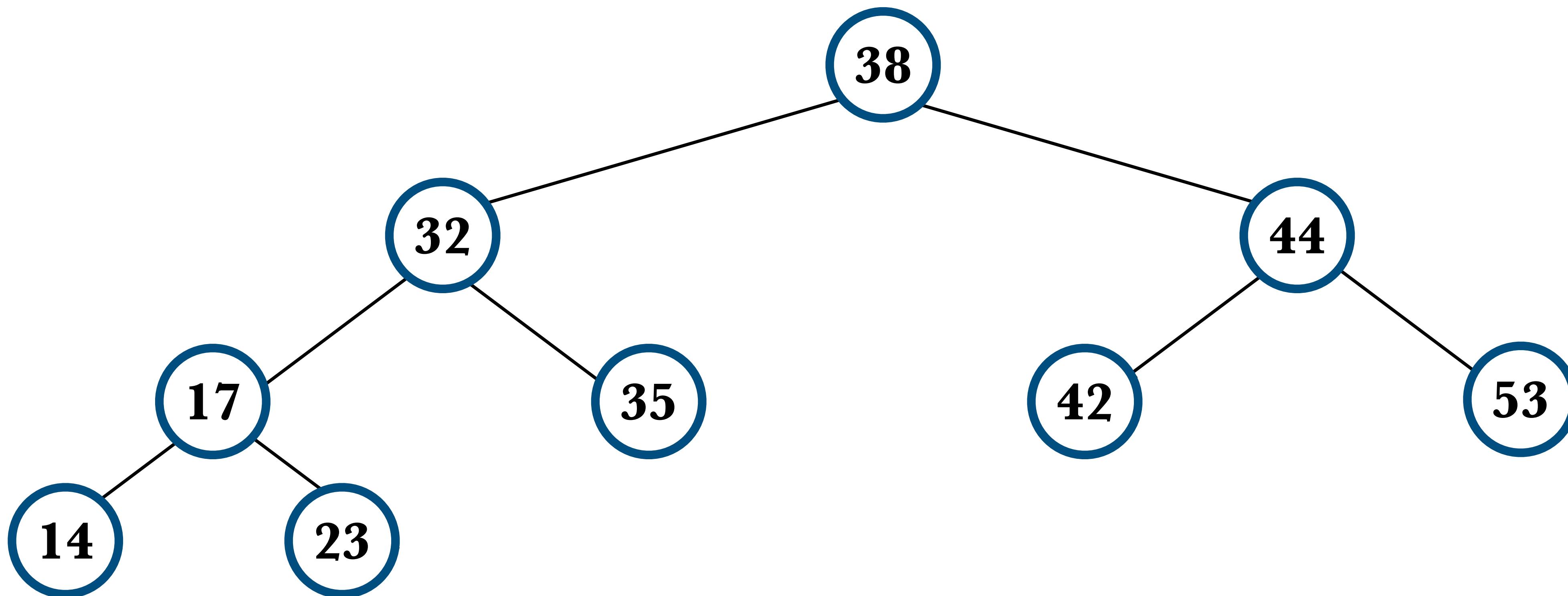
Xóa khỏi cây T nút **40**:



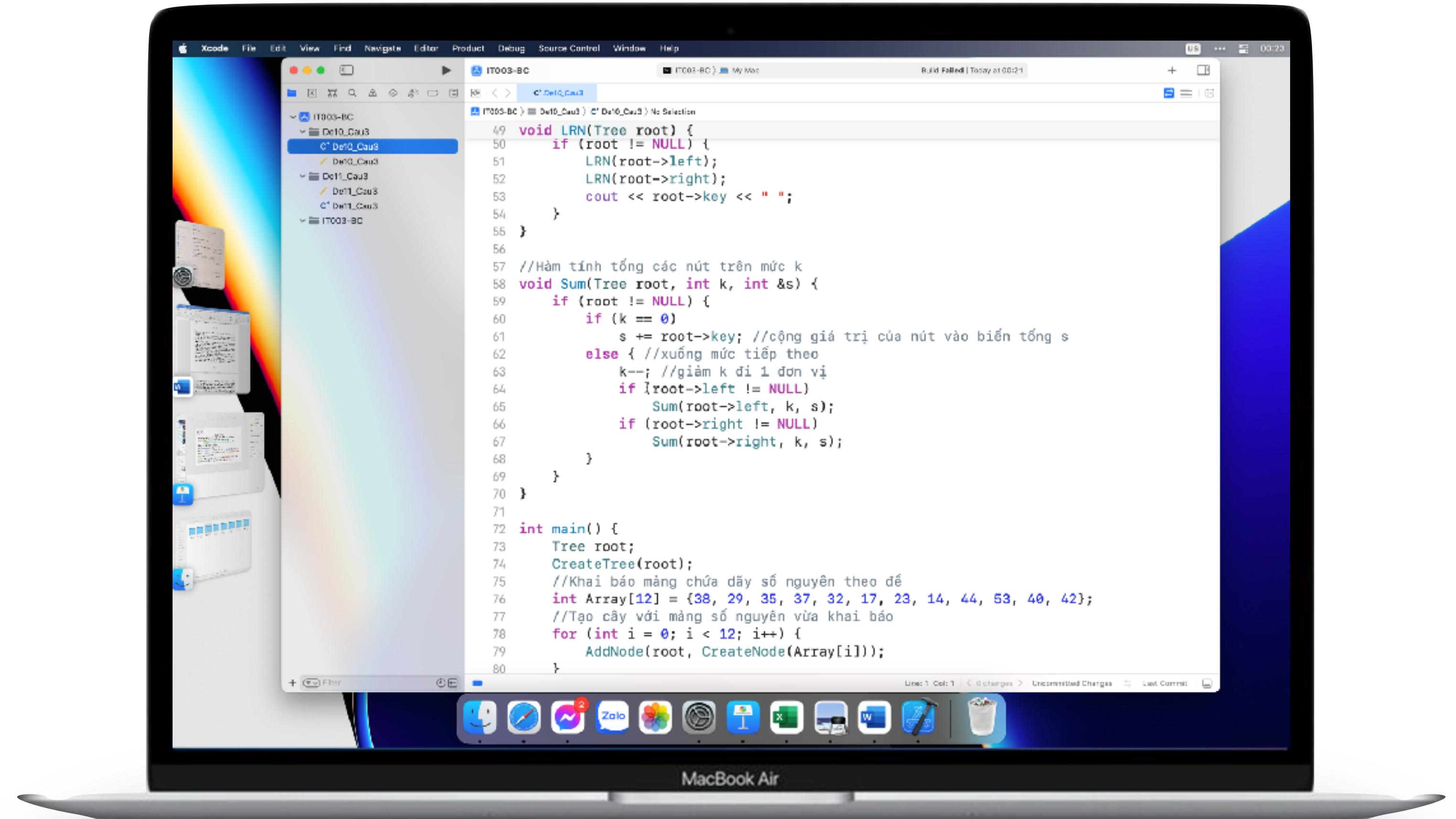
Xóa khỏi cây T nút 29:



Xóa khỏi cây T nút 38:



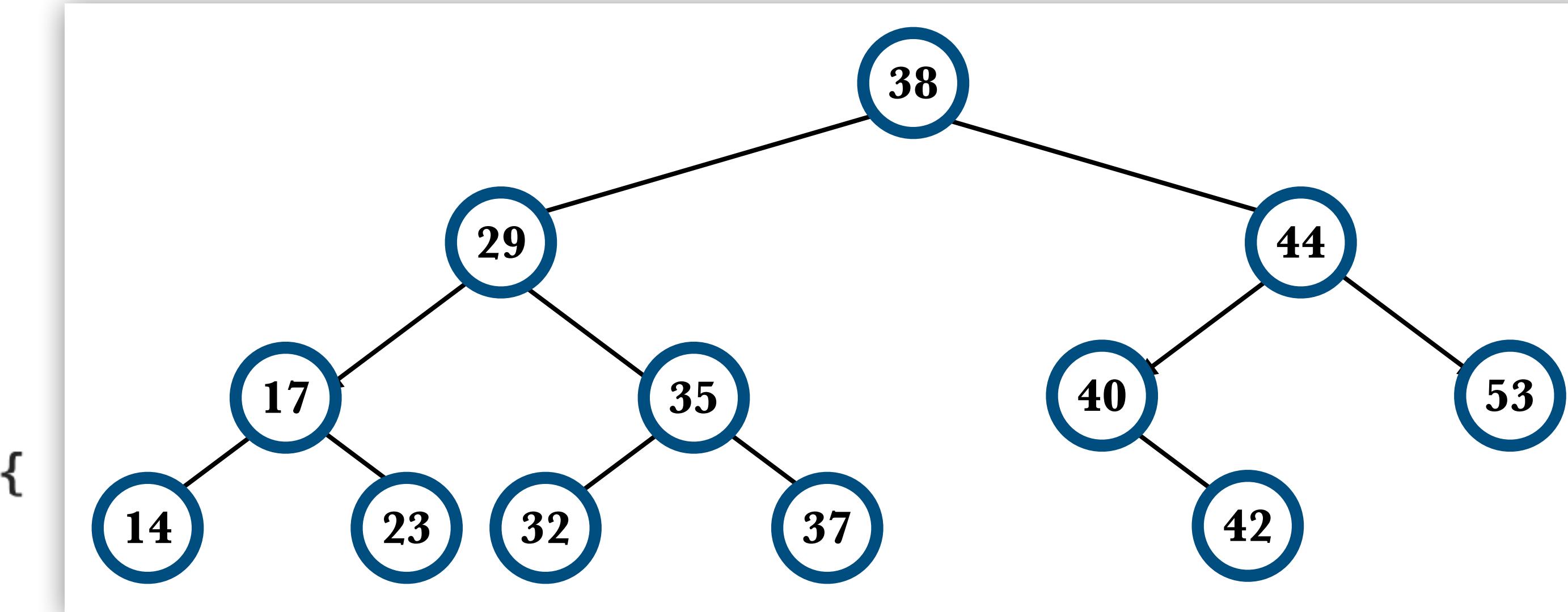
c. Viết hàm in ra **tổng giá trị các nút ở mức k** trên cây nhị phân tìm kiếm T (với quy ước mức ở nút gốc là 0).



```

51 LRN(root->left);
52 LRN(root->right);
53 cout << root->key << " ";
54 }
55
56 //Hàm tính tổng các nút trên mức k
57 void Sum(Tree root, int k, int &s) {
58     if (root != NULL) {
59         if (k == 0)
60             s += root->key; //cộng giá trị của nút vào biến tổng s
61     else { //xuống mức tiếp theo
62         k--; //giảm k đi 1 đơn vị
63         if (root->left != NULL)
64             Sum(root->left, k, s);
65         if (root->right != NULL)
66             Sum(root->right, k, s);
67     }
68 }
69 }
70 }
71
72 int main() {
73     Tree root;
74     CreateTree(root);

```



Nhập mức k bạn muốn tính tổng các node: 2

Kết quả: 145

Program ended with exit code: 0

CÂY NHỊ PHÂN TÌM KIẾM

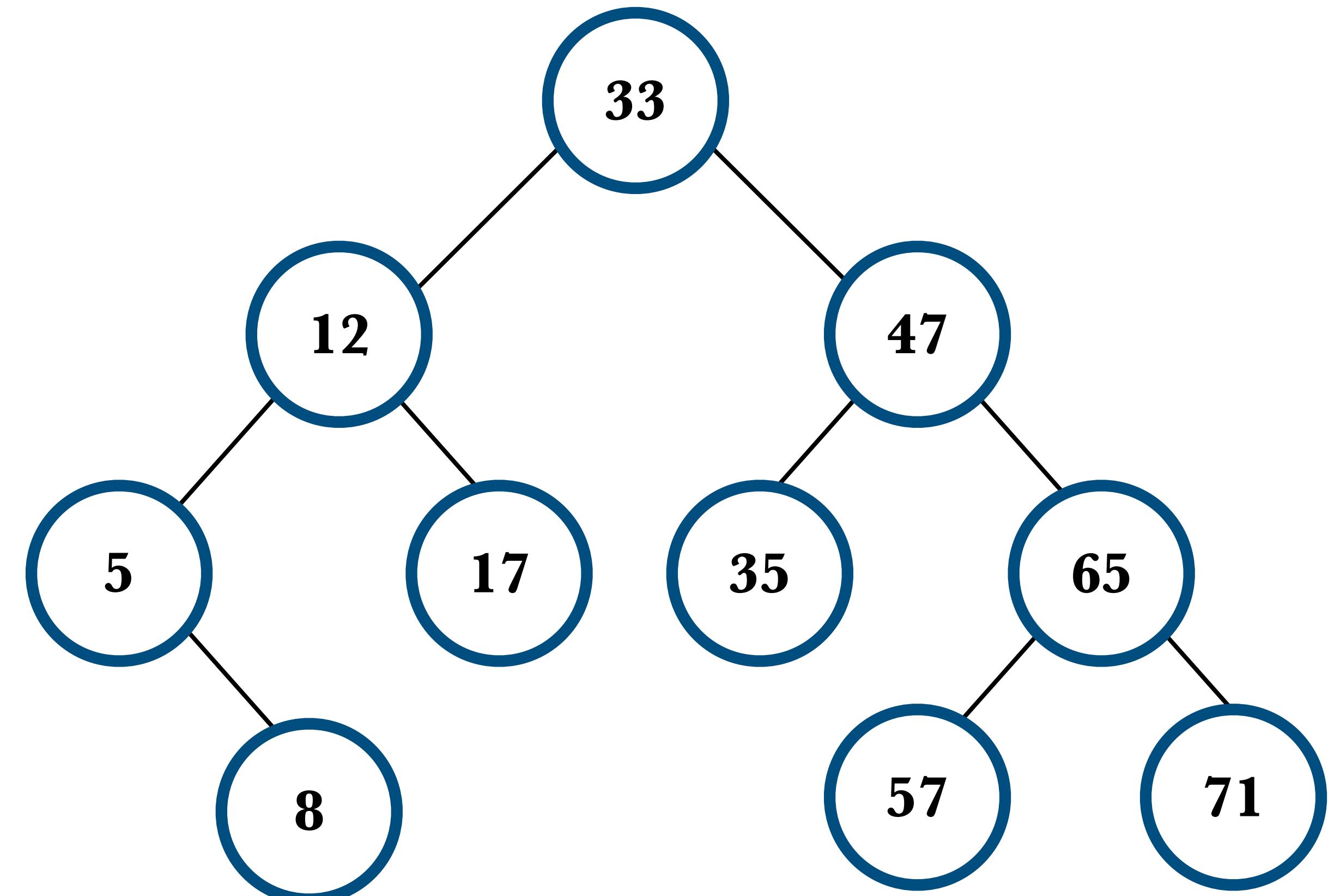
Đề 11: Câu 3

ĐỀ BÀI

Đề 11: Câu 3

Cho **cây nhị phân tìm kiếm T** như hình bên. Anh/chị hãy thực hiện:

- a.** Viết hàm **in ra các nút trên cây tại mức thứ k** của cây. Ví dụ: nhập $k = 0$ (mức gốc) thì in ra 33.
- b.** Hãy vẽ lại cây khi **xóa lần lượt nút 33; 35; 12; 5**.



a. Viết hàm **in ra các nút trên cây tại mức thứ k** của cây. Ví dụ: nhập k = 0 (mức gốc) thì in ra 33.

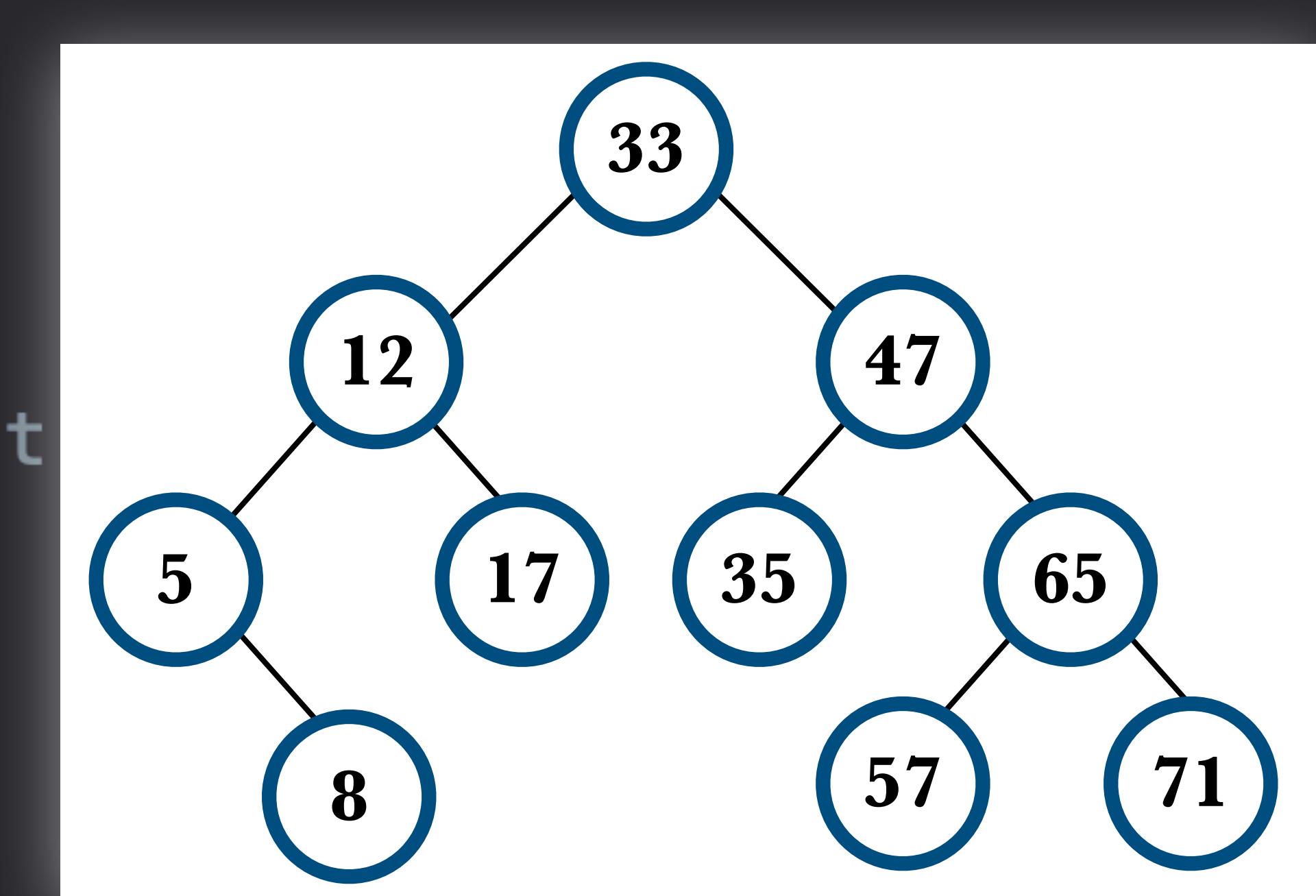
```
57 //Hàm in các node trên mức k
58 void Print_Nodes(Tree root, int k) {
59     if (root != NULL) {
60         if (k == 0)
61             cout << root->key << " ";
62         else { //xuống mức tiếp theo
63             k--;
64             if (root->left != NULL)
65                 Print_Nodes(root->left, k);
66             if (root->right != NULL)
67                 Print_Nodes(root->right, k);
68         }
69     }
70 }
71
72 int main() {
73     Tree root;
74     CreateTree(root);
75     //Khai báo mảng chứa các nút số nguyên trong cây
76     int Array[12] = {33, 12, 47, 5, 17, 35, 65, 8, 57, 71};
77     //Tạo cây với mảng số nguyên vừa khai báo
78     for (int i = 0; i < 10; i++) {
79         AddNode(root, CreateNode(Array[i]));
80     }
81     cout << "Nhập mức k bạn muốn in các node: ";
82     int k;
83     cin >> k;
84     cout << "Kết quả: " ;
85     Print_Nodes(root, k);
86     return 0;
87 }
```

```

//Hàm in các node trên mức k
void Print_Nodes(Tree root, int k) {
    if (root != NULL) {
        if (k == 0)
            cout << root->key << " "; //in nút
        else { //xuống mức tiếp theo
            k--; //giảm k đi 1 đơn vị
            if (root->left != NULL)
                Print_Nodes(root->left, k);
            if (root->right != NULL)
                Print_Nodes(root->right, k);
        }
    }
}

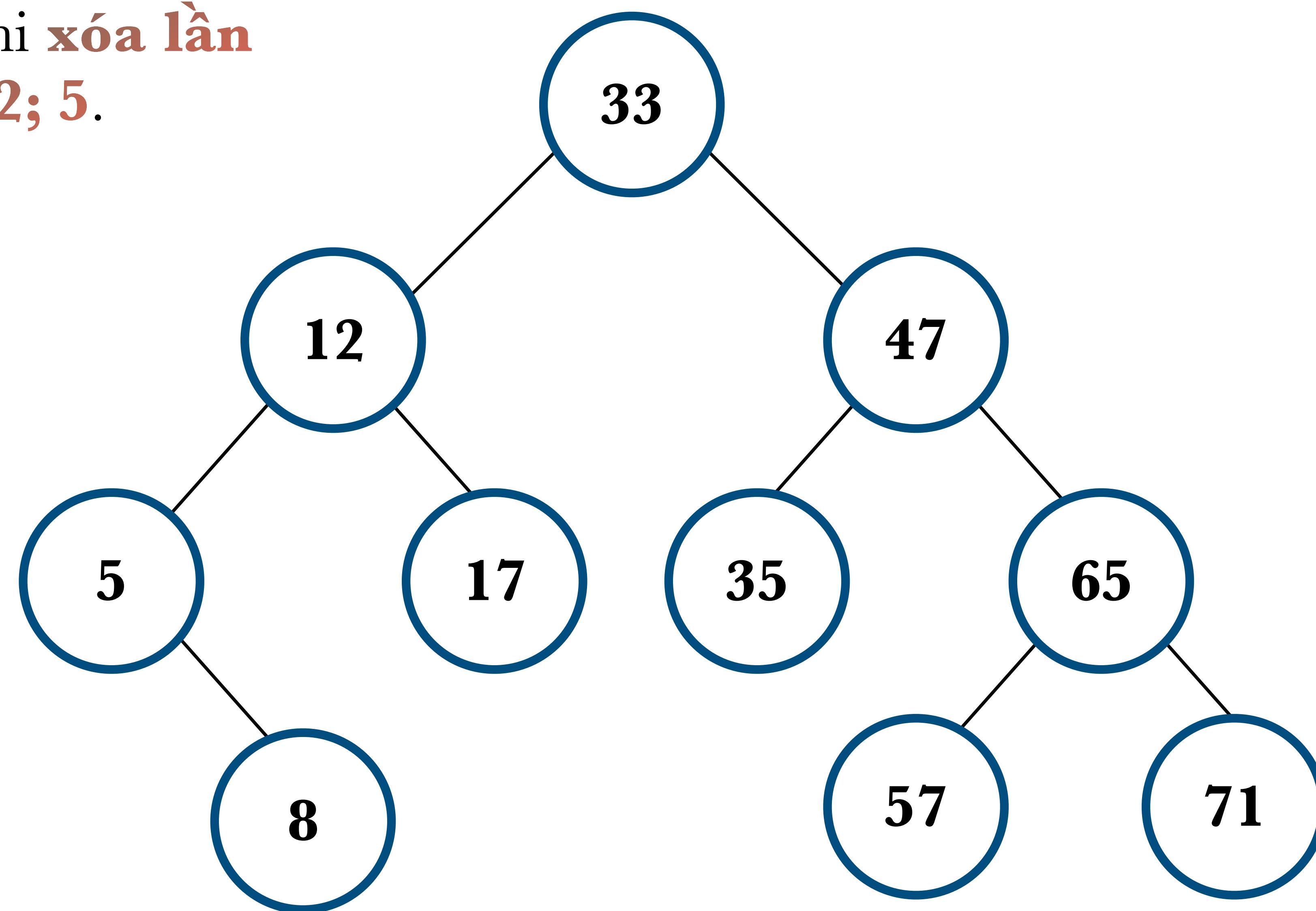
int main() {
    Tree root;
    CreateTree(root);
    //Khai báo mảng chứa các nút số nguyên trong cây
    int Array[12] = {33, 12, 47, 5, 17, 35, 65, 8, 57, 71};
}

```

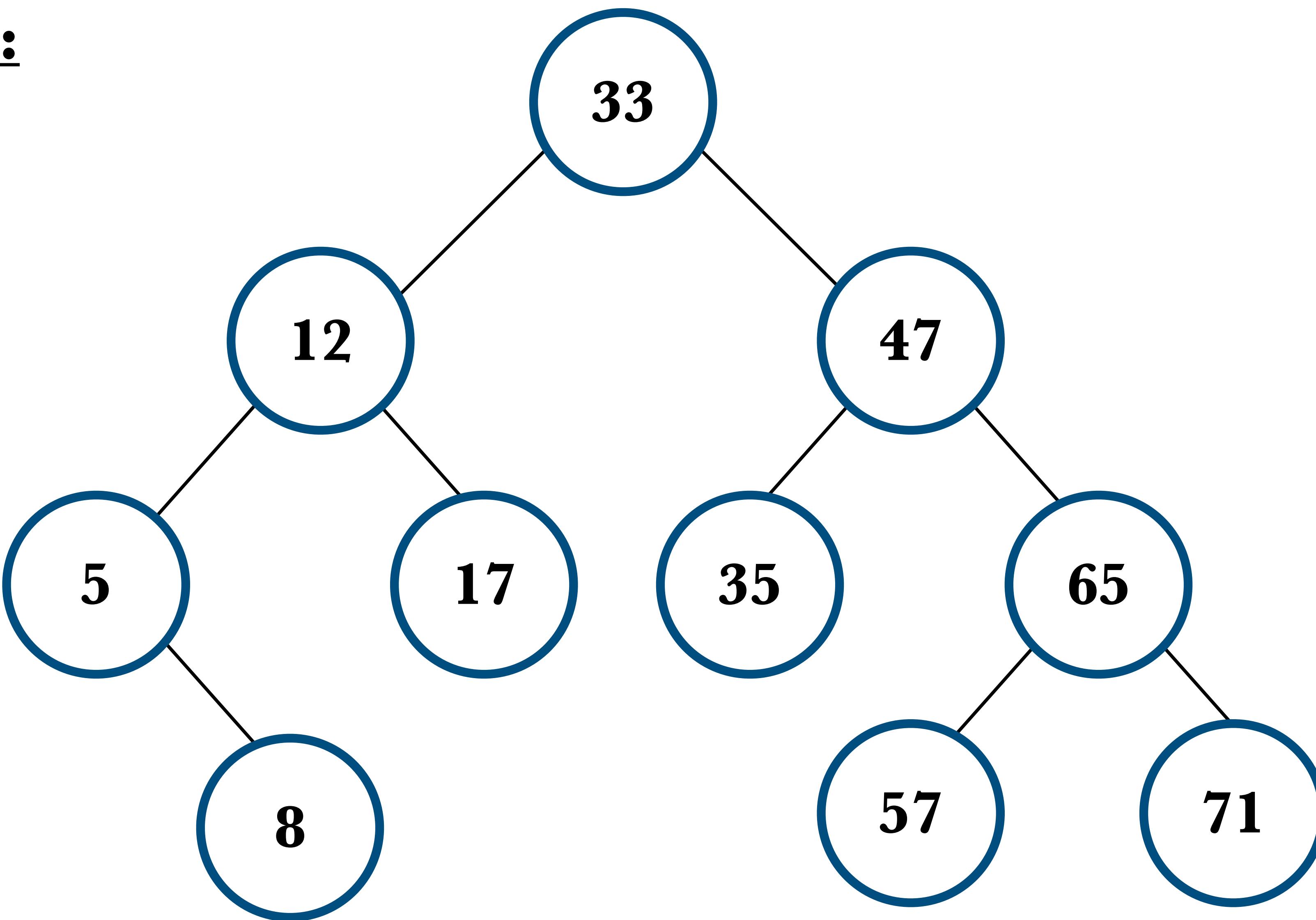


Nhập mức k bạn muốn in các node: 3
 Kết quả: 8 57 71 Program ended with exit code: 0

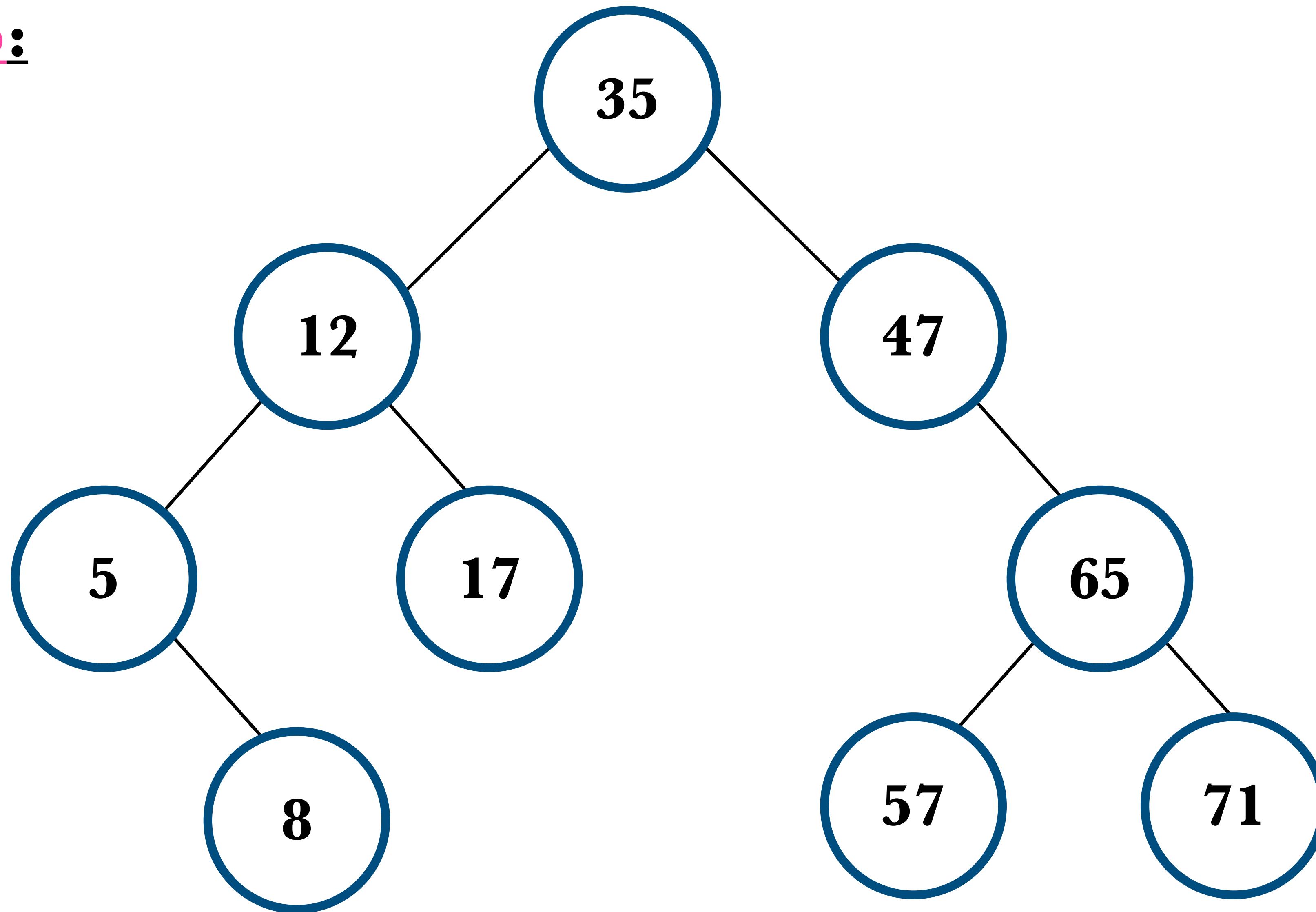
b. Hãy vẽ lại cây khi **xóa lần lượt** nút 33; 35; 12; 5.



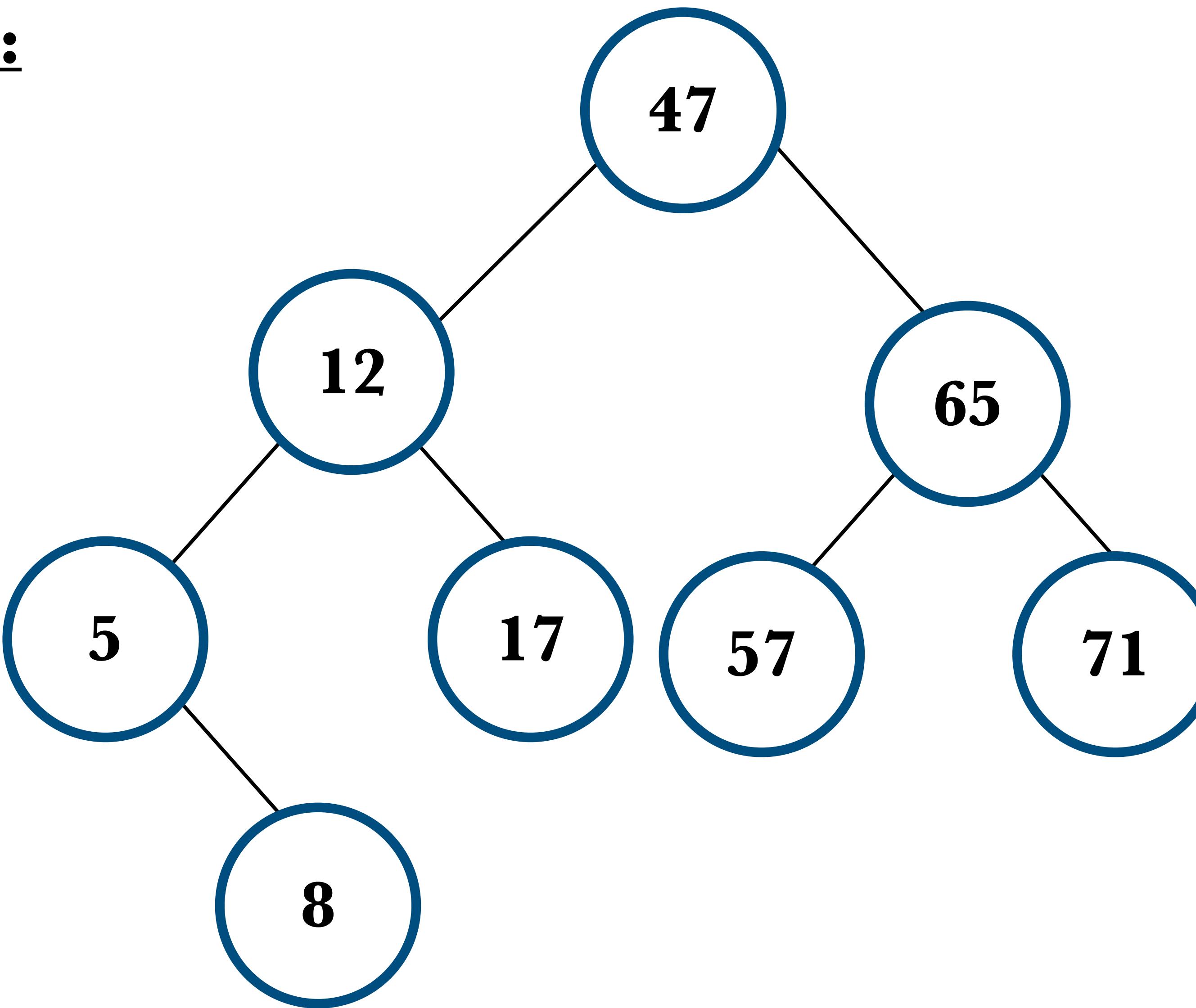
Xóa khỏi cây T nút 33:



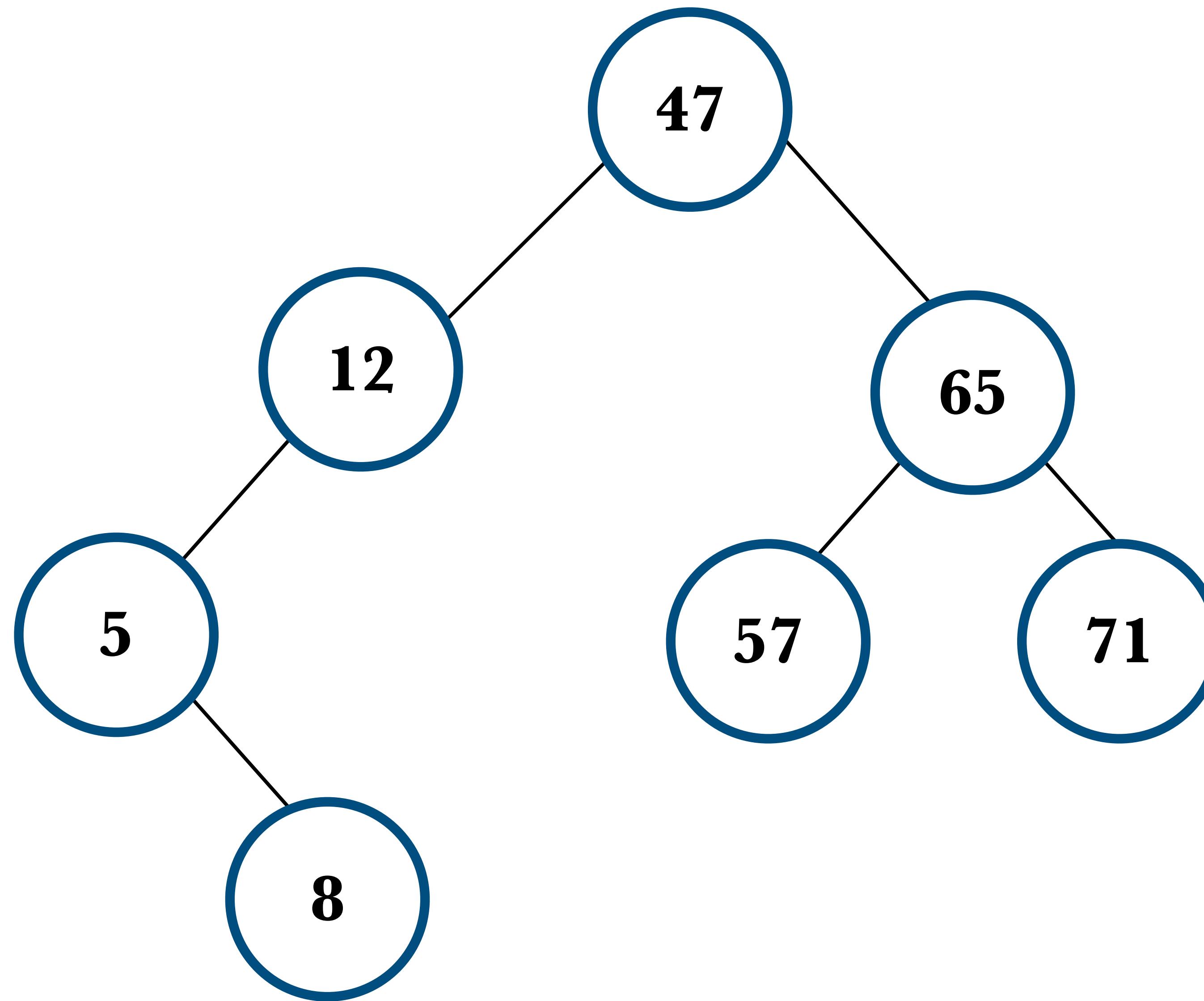
Xóa khỏi cây T nút 35:



Xóa khỏi cây T nút 12:



Xóa khỏi cây T nút 5:



TỔNG KẾT

IT003.N27.1 - GROUP 4 - CODE IV

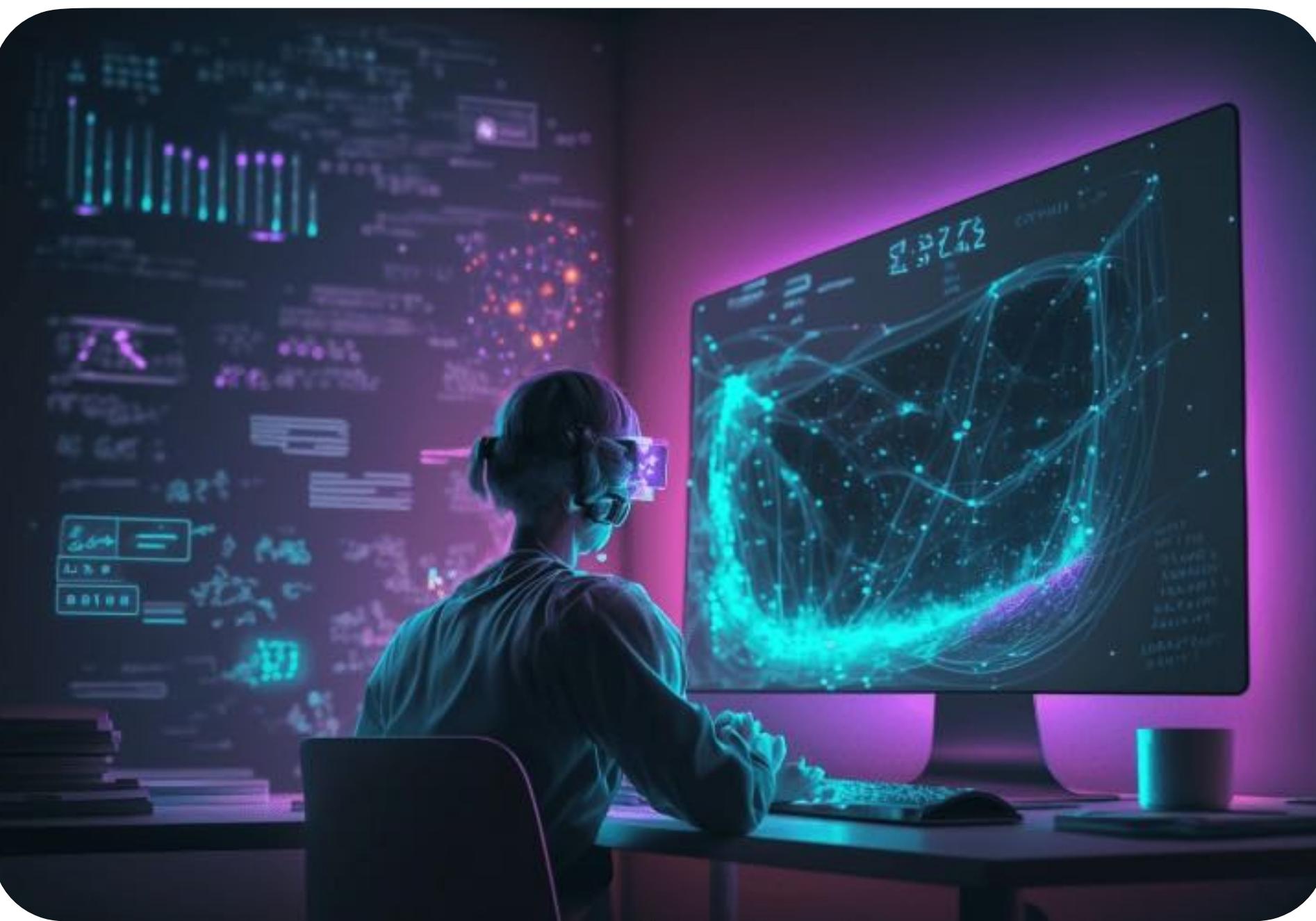
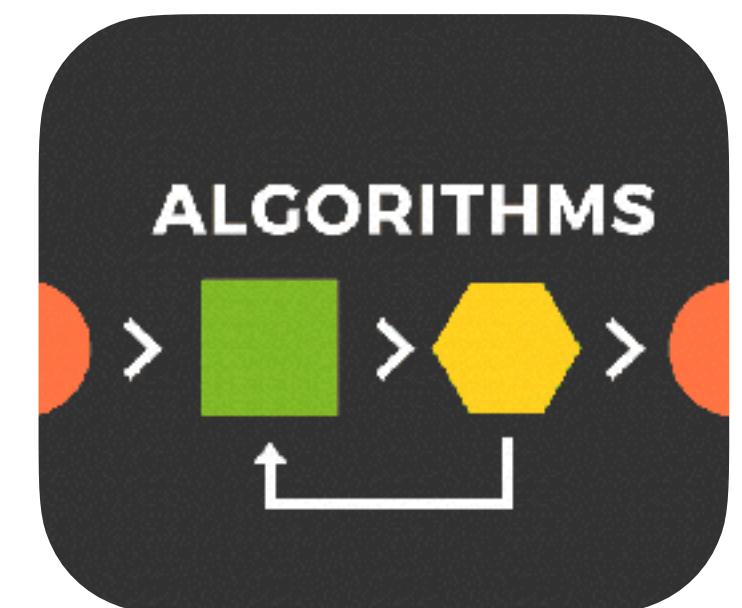
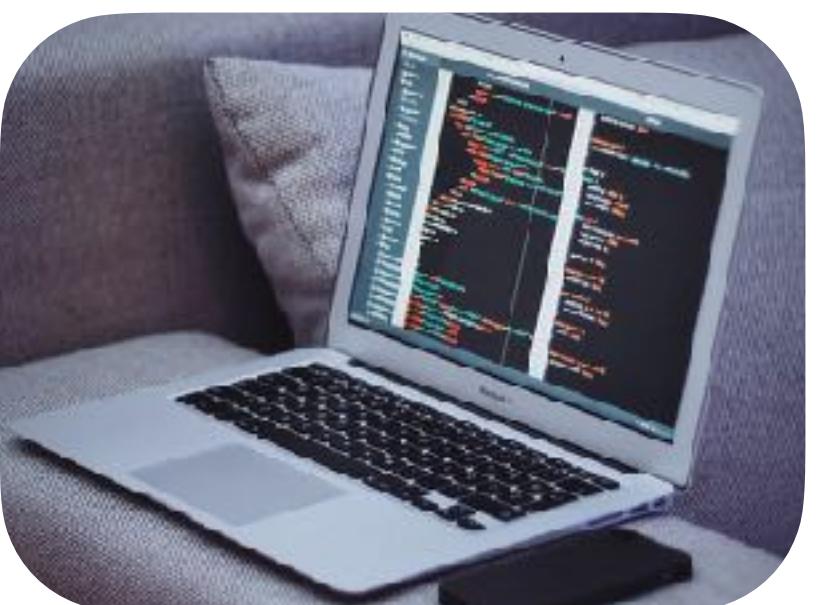


THUẬT TOÁN CHỌN TRỰC TIẾP
(SELECTION SORT)

Ý TƯỞNG CỦA THUẬT
TOÁN TÌM KIẾM NHỊ PHÂN

(BINARY SEARCH)

ĐỊNH NGHĨA
CÂY NHỊ PHÂN TÌM KIẾM.



ÔN TẬP
CÂY NHỊ PHÂN
TÌM KIẾM
HIỂU ĐƯỢC
XOÁ NÚT TRÊN
CÂY

CẤU TRÚC VÀ NHỮNG HÀM LIÊN QUAN ĐẾN

DANH SÁCH LIÊN KẾT

ƯU ĐIỂM BẢNG BĂM

KHI SO VỚI CẤU TRÚC DỮ LIỆU ĐÃ HỌC

