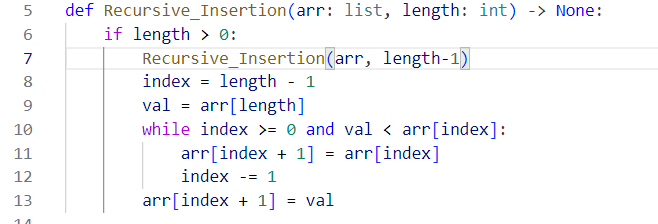
DESIGN AND ANALYSIS OF ALGORITHMS

**Ex1**

****

1/ basic operation: val < arr[index] on line 13

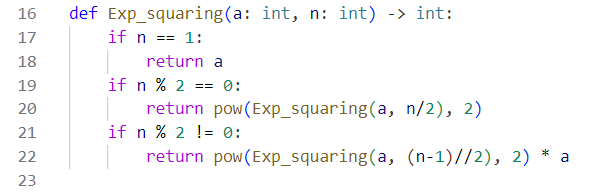
2/ Worst case: Input A is sorted in decreasing order

3/Counting the number of basic operations in the worst case:

**Time efficiency**

***Cworst*(*n*) = *n*(*n*-1)/2 ∈ Θ(*n*2)**

**Ex2**

****

1/ basic operation: n % 2 == 0 on line 19

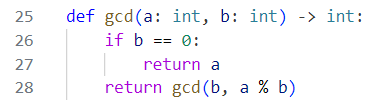
2/ Worst case: n is large number

3/Counting the number of basic operations in the worst case:

**Time efficiency**

***Cworst*(*n*) = *O(logn)∈Θ(logn)***

**Ex3**



1/ basic operation: return gcd(b, a % b) on line 28

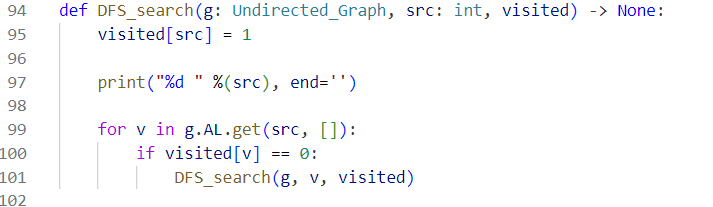
2/ Worst case: a, b is large number

3/Counting the number of basic operations in the worst case:

**Time efficiency**

***Cworst*(*n*) = *O(logn)∈Θ(logn)***

**Ex4**

****

1/ basic operation: (visited[src] = 1 and if visited[v] == 0)

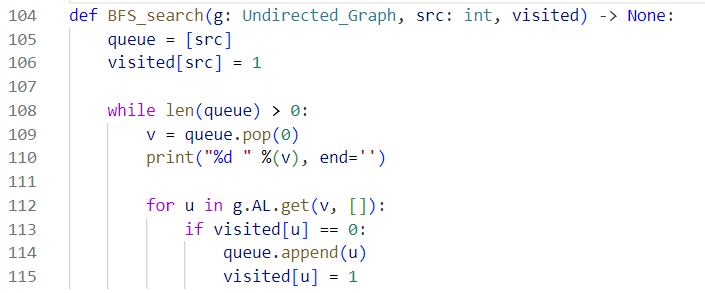
2/ Worst case: Complete graph

3/Counting the number of basic operations in the worst case:

**Time efficiency**

***Cworst*(*n*) = *O(V+E)***

**Ex5**

****

1/ basic operation: (visited[src] = 1 and if visited[v] == 0)

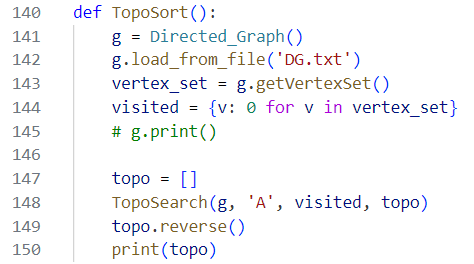
2/ Worst case: Complete graph

3/Counting the number of basic operations in the worst case:

**Time efficiency**

***Cworst*(*n*) = *O(V+E)***

**Ex6**

****

1/ basic operation: (visited[src] = 1 and if visited[v] == 0)

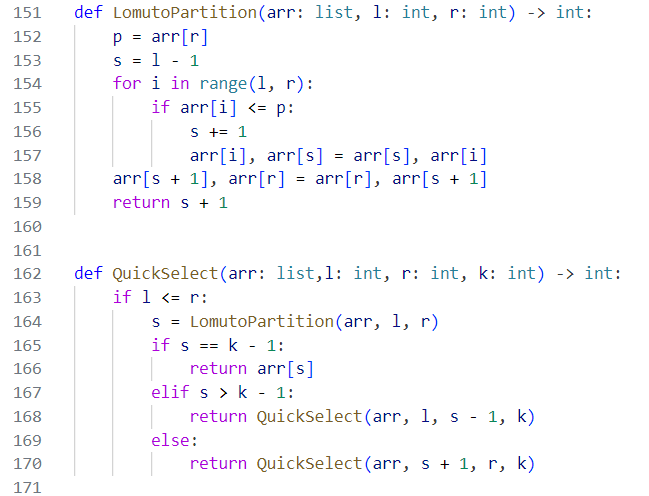
2/ Worst case: Complete graph

3/Counting the number of basic operations in the worst case:

**Time efficiency**

***Cworst*(*n*) = *O(V+E)***

**Ex7:**

****

1/ basic operation: arr[i] <= p, arr[i], arr[s] = arr[s], arr[i]

2/ Worst case:

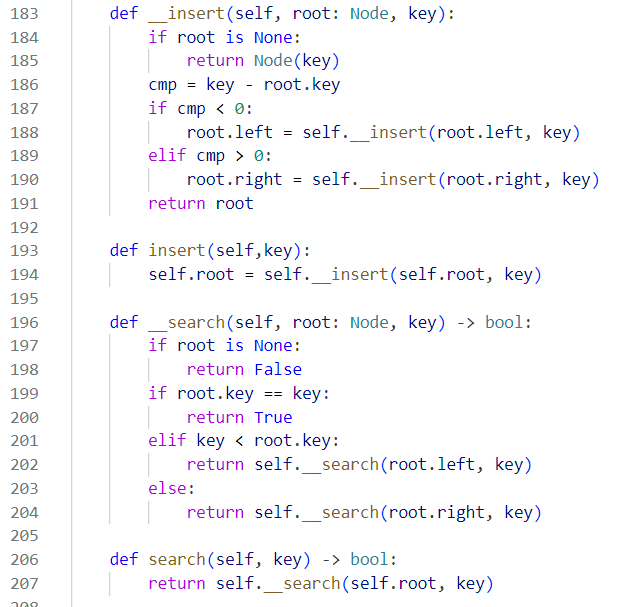
* Trường hợp xấu nhất cho LomutoPartition xảy ra khi mốc (pivot) luôn nằm ở đầu hoặc cuối của mảng phân hoạch, dẫn đến phân hoạch không đều. Điều này xảy ra khi mảng đã sắp xếp hoàn toàn hoặc đảo ngược, khiến số lần phân hoạch tối đa gần bằng số phần tử trong mảng.
* Trường hợp xấu nhất của QuickSelect xảy ra khi phải phân hoạch toàn bộ mảng mà không loại bỏ được nhiều phần tử trong mỗi lần gọi đệ quy.

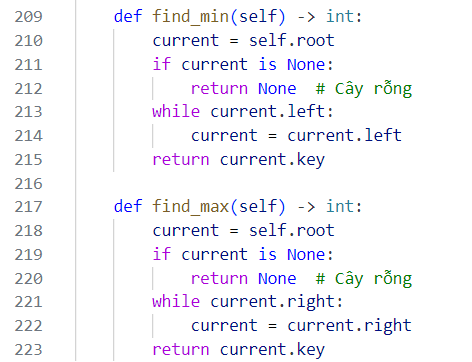
3/Counting the number of basic operations in the worst case:

**Time efficiency**

***Cworst*(*n*) = *O(n^2)***

**Ex8**

****

****

1/ basic operation:

* Trong hàm \_\_insert, phép toán cơ bản là so sánh giá trị key với root.key (cmp = key - root.key) và cập nhật con trái hoặc con phải (root.left = ..., root.right = ...).
* Trong hàm \_\_search, phép toán cơ bản là kiểm tra giá trị root.key == key và điều hướng sang con trái hoặc con phải của cây.
* Trong các hàm find\_min và find\_max, phép toán cơ bản là duyệt lần lượt xuống con trái (để tìm giá trị nhỏ nhất) hoặc con phải (để tìm giá trị lớn nhất).

2/ Worst case: tree as linkedlist

3/Counting the number of basic operations in the worst case:

**Time efficiency**

***Cworst*(*n*) = *O(n)***