Nama : Alvilio Daras

NIM : 18220011

1. Jawaban: Space Complexity dari program tersebut adalah O(n), karena membutuhkan array dengan ukuran n.
2. Jawaban: Space Complexity dari program tersebut adalah O(1), karena hanya menggunakan satu variabel tambahan.
3. Jawaban: Space Complexity dari program tersebut adalah O(n), karena membutuhkan stack untuk setiap pemanggilan rekursif yang berukuran n.
4. Analisis Space Complexity dari algoritma pencarian biner:

* Kompleksitas waktu algoritma pencarian biner adalah O(log n), di mana n adalah ukuran array yang dicari.
* Kompleksitas ruang algoritma pencarian biner dapat dianggap O(1), yang berarti bahwa algoritma hanya membutuhkan sedikit ruang tambahan selain ruang yang dibutuhkan untuk menyimpan array.
* Namun, kompleksitas ruang dapat meningkat jika algoritma menciptakan variabel tambahan atau melakukan tugas tambahan selain pencarian biner.

1. Analisis Space Complexity dari algoritma insertion sort:

* Algoritma insertion sort melakukan pengurutan dengan memindahkan elemen array ke posisi yang tepat, satu per satu, hingga seluruh array terurut.
* Algoritma insertion sort hanya membutuhkan ruang tambahan konstan untuk memori yang digunakan oleh variabel iterasi dan beberapa variabel bantu lainnya.
* Oleh karena itu, kompleksitas ruang algoritma insertion sort dapat dianggap O(1), yang berarti bahwa ruang tambahan yang dibutuhkan tidak tumbuh seiring bertambahnya ukuran array yang diurutkan.
* Kompleksitas ruang algoritma insertion sort tetap konstan, tidak tergantung pada ukuran array yang diurutkan, sehingga algoritma ini memerlukan sedikit ruang tambahan.

1. Analisis Space Complexity dari algoritma quick sort:

* Algoritma quick sort menggunakan pendekatan divide and conquer untuk mengurutkan array dengan membaginya menjadi dua bagian dan mengurutkan masing-masing bagian secara terpisah.
* Untuk melakukan ini, algoritma quick sort menggunakan teknik rekursi yang membutuhkan tambahan ruang untuk memanggil fungsi rekursif dan menyimpan variabel lokal di setiap tingkat panggilan rekursif.
* Oleh karena itu, kompleksitas ruang algoritma quick sort bergantung pada kedalaman panggilan rekursif dan jumlah variabel lokal yang digunakan di setiap tingkat panggilan.
* Kompleksitas ruang algoritma quick sort dapat mencapai O(n) dalam kasus terburuk ketika panggilan rekursif mencapai kedalaman n, di mana n adalah ukuran array yang diurutkan. Namun, pada umumnya kompleksitas ruang quick sort dapat dianggap O(log n) dalam kasus rata-rata.
* Jumlah ruang tambahan yang dibutuhkan oleh algoritma quick sort dapat diminimalkan dengan teknik seperti pengurangan rekursif dan penggunaan iterasi alih-alih rekursi, yang dapat membatasi kedalaman panggilan rekursif dan mengurangi penggunaan variabel lokal.