Analisis Algoritma

A. Bubble sort dan Insertio sort

1. Bubble sort
    a. Worst Case: Dalam kasus terburuk, Bubble Sort memiliki kompleksitas waktu O(n^2). Ini terjadi ketika array input sudah terurut secara terbalik. Dalam hal ini, Bubble Sort akan melakukan iterasi penuh pada setiap pasangan elemen dan menukar posisi mereka. Sehingga jumlah langkah yang dibutuhkan menjadi maksimum.
    b. Best Case: Dalam kasus terbaik, Bubble Sort memiliki kompleksitas waktu O(n). Ini terjadi ketika array input sudah terurut dengan benar, sehingga tidak ada pertukaran yang perlu dilakukan selama iterasi. Meskipun demikian, dalam implementasi di atas, Bubble Sort tetap melakukan semua perbandingan.
    c. Average Case: Dalam kasus rata-rata, Bubble Sort memiliki kompleksitas waktu O(n^2). Ini terjadi ketika elemen-elemen array input secara acak berada dalam urutan yang tidak teratur. Bubble Sort akan melakukan beberapa iterasi dan pertukaran untuk mengurutkan array secara keseluruhan.
2. Insertion Sort:
    a. Worst Case: Dalam kasus terburuk, Insertion Sort memiliki kompleksitas waktu O(n^2). Ini terjadi ketika array input sudah terurut secara terbalik. Pada setiap iterasi, setiap elemen akan harus digeser ke posisi yang benar, sehingga membutuhkan banyak pergeseran elemen.
    b. Best Case: Dalam kasus terbaik, Insertion Sort memiliki kompleksitas waktu O(n). Ini terjadi ketika array input sudah terurut dengan benar. Dalam hal ini, Insertion Sort hanya memerlukan satu iterasi melalui elemen-elemen array tanpa melakukan pergeseran.
    c. Average Case: Dalam kasus rata-rata, Insertion Sort memiliki kompleksitas waktu O(n^2). Ini terjadi ketika elemen-elemen array input secara acak berada dalam urutan yang tidak teratur. Insertion Sort akan

melakukan beberapa pergeseran elemen untuk mengurutkan array secara keseluruhan.

B. TSP dan Djikstra

1. Algoritma TSP:
    a. Worst Case: Algoritma TSP yang digunakan pada kode program di atas adalah brute force, di mana semua kemungkinan jalur akan dijelajahi. Dalam kasus ini, kompleksitas waktu algoritma TSP adalah O(n!), di mana n adalah jumlah simpul dalam grafik. Jadi, dalam kasus terburuk, dengan peningkatan jumlah simpul, waktu eksekusi algoritma TSP secara eksponensial meningkat.
    b. Best Case: Algoritma TSP brute force tidak memiliki kasus terbaik yang secara signifikan. Dalam kode program di atas, setiap kali algoritma TSP dipanggil, simpul awal yang tetap sama ('A') dan seluruh grafik akan dieksplorasi. Sehingga, kompleksitas waktu algoritma TSP akan tetap menjadi O(n!) dalam kasus terbaik.
    c. Average Case: Algoritma TSP brute force tidak memiliki analisis yang jelas untuk kasus rata-rata. Kompleksitas waktu yang tinggi, O(n!), membuatnya tidak efisien dalam menghitung jalur terpendek pada grafik yang lebih besar.
2. Algoritma Dijkstra:
    a. Worst Case: Algoritma Dijkstra digunakan untuk menghitung jalur terpendek antara dua simpul. Dalam kasus terburuk, ketika semua simpul harus dijelajahi, kompleksitas waktu algoritma Dijkstra adalah O((V + E) log V), di mana V adalah jumlah simpul dan E adalah jumlah tepi dalam grafik. Dalam kode program di atas, algoritma Dijkstra digunakan untuk menghitung jalur terpendek antara dua simpul yang diberikan oleh pengguna. Jadi, dalam kasus terburuk, kompleksitas waktu algoritma Dijkstra tergantung pada jumlah simpul dan tepi dalam grafik yang diberikan.
    b. Best Case: Kasus terbaik algoritma Dijkstra terjadi ketika simpul awal dan akhir sama, sehingga tidak perlu menjelajahi grafik lebih jauh.

Kompleksitas waktu dalam kasus terbaik adalah O(1), di mana algoritma Dijkstra dapat langsung mengembalikan jalur terpendek yang sama simpul awal dan akhir.

c. Average Case: Dalam kasus rata-rata, kompleksitas waktu algoritma Dijkstra adalah O((V + E) log V), di mana V adalah jumlah simpul dan E adalah jumlah tepi dalam grafik. Hal ini didasarkan pada asumsi bahwa grafik tidak memiliki karakteristik khusus yang mempengaruhi kinerja algoritma. Dalam praktiknya, algoritma Dijkstra dikenal memiliki kinerja yang baik pada grafik dengan jumlah simpul dan tepi yang moderat.