

TUGAS MANDIRI
PERANCANGAN DAN ANALISIS ALGORITMA



DOSEN PENGAMPU:
Randi Proska Sandra, M.Sc

OLEH:
Rayhan Ahadi Nifri
21343035
Informatika

PROGRAM STUDI INFORMATIKA
DEPARTEMEN ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG
2023

A. Dynamic Programing Warshall's and Floyd's Algorithms

Warshall's dan Floyd's Algorithms adalah dua algoritma yang digunakan untuk menghitung penutup transitif dari sebuah digraf. Penutup transitif adalah matriks yang menunjukkan apakah ada jalur terarah antara setiap pasang vertex dalam digraf.

Warshall's Algorithm adalah algoritma yang menggunakan pendekatan rekursif untuk menghitung penutup transitif dalam waktu $O(n^3)$, di mana n adalah jumlah vertex dalam digraf. Algoritma ini bekerja dengan menggunakan sebuah matriks A untuk merepresentasikan graf awal, dan menggunakan teknik dinamis untuk memperbarui matriks tersebut secara iteratif hingga mendapatkan penutup transitif. Algoritma ini cukup efisien untuk digunakan pada graf dengan jumlah vertex yang tidak terlalu besar.

Floyd's Algorithm, juga dikenal sebagai algoritma Warshall–Floyd, adalah algoritma yang menggunakan pendekatan iteratif untuk menghitung penutup transitif dalam waktu $O(n^3)$. Algoritma ini menggunakan matriks A untuk merepresentasikan graf awal, dan menggunakan teknik dinamis untuk memperbarui matriks tersebut secara iteratif hingga mendapatkan penutup transitif. Namun, algoritma ini lebih efisien daripada Warshall's Algorithm karena menggunakan teknik pemrograman dinamis yang lebih efisien. Selain itu, Floyd's Algorithm dapat digunakan untuk mencari jalur terpendek antara setiap pasang vertex dalam digraf.

Dalam kedua algoritma ini, matriks yang diperbarui selama proses adalah matriks penutup transitif yang akan menyimpan informasi tentang keberadaan jalur terarah antara setiap pasang vertex dalam digraf. Kedua algoritma ini memiliki kompleksitas waktu yang sama, tetapi Floyd's Algorithm lebih efisien dalam hal penggunaan memori dan dapat digunakan untuk mencari jalur terpendek antara setiap pasang vertex dalam digraf.

B. PSEUDOCODE

Warshall's Algorithms

Input: Matriks A berukuran $n \times n$ yang merepresentasikan digraf

Output: Matriks penutup transitif T dari digraf

Untuk $i = 1$ hingga n , lakukan:

 Untuk $j = 1$ hingga n , lakukan:

$T[i][j] = A[i][j]$

Untuk $k = 1$ hingga n , lakukan:

 Untuk $i = 1$ hingga n , lakukan:

 Untuk $j = 1$ hingga n , lakukan:

$T[i][j] = T[i][j] \text{ OR } (T[i][k] \text{ AND } T[k][j])$

Output T

Floyd's Algorithms

Input: Matriks A berukuran $n \times n$ yang merepresentasikan digraf

Output: Matriks penutup transitif T dari digraf

Untuk $i = 1$ hingga n , lakukan:

 Untuk $j = 1$ hingga n , lakukan:

$T[i][j] = A[i][j]$

Untuk $k = 1$ hingga n , lakukan:

 Untuk $i = 1$ hingga n , lakukan:

 Untuk $j = 1$ hingga n , lakukan:

$T[i][j] = \text{MIN}(T[i][j], T[i][k] + T[k][j])$

Output T

Kedua algoritma ini memiliki struktur yang mirip, yaitu menggunakan teknik dinamis untuk memperbarui matriks T secara iteratif hingga mendapatkan penutup transitif. Perbedaannya terletak pada cara memperbarui matriks T. Warshall's Algorithm menggunakan operasi logika OR dan AND, sedangkan Floyd's Algorithm menggunakan operasi matematika MIN

C. SOURCE CODE

```
1  # Warshall's Algorithm
2  def warshall_algorithm(graph):
3      n = len(graph)
4      T = [[0 for j in range(n)] for i in range(n)]
5
6      for i in range(n):
7          for j in range(n):
8              T[i][j] = graph[i][j]
9
10     for k in range(n):
11         for i in range(n):
12             for j in range(n):
13                 T[i][j] = T[i][j] or (T[i][k] and T[k][j])
14
15     return T
16
17 # Floyd's Algorithm
18 def floyd_algorithm(graph):
19     n = len(graph)
20     T = [[0 for j in range(n)] for i in range(n)]
21
22     for i in range(n):
23         for j in range(n):
24             T[i][j] = graph[i][j]
25
26     for k in range(n):
27         for i in range(n):
28             for j in range(n):
29                 T[i][j] = min(T[i][j], T[i][k] + T[k][j])
30
31     return T
```

Kedua algoritma ini menerima matriks representasi graf sebagai input dan mengembalikan matriks penutup transitif dari graf tersebut. Matriks penutup transitif disimpan dalam variabel T yang diinisialisasi dengan matriks representasi graf awal. Setelah itu, dilakukan iterasi untuk memperbarui matriks T hingga mendapatkan penutup transitif menggunakan teknik dinamis yang telah dijelaskan pada penjelasan dan pseudocode sebelumnya.

D. ANALISA KEBUTUHAN WAKTU

Materi di atas membahas tentang Warshall's Algorithm dan Floyd's Algorithm, dua algoritma yang digunakan untuk mencari penutup transitif dari suatu graf. Kedua algoritma ini dapat digunakan untuk menyelesaikan berbagai masalah yang terkait dengan penentuan jarak atau hubungan antar simpul pada sebuah graf. Untuk menganalisa kebutuhan waktu materi di atas, tergantung pada kompleksitas graf yang akan dihitung penutup transitifnya. Untuk graf yang kecil, kedua algoritma ini dapat menyelesaikan perhitungan dengan cepat dan dalam waktu yang relatif singkat. Namun, untuk graf yang besar dengan jumlah simpul dan sisi yang banyak, kedua algoritma ini dapat memerlukan waktu yang cukup lama untuk menyelesaikan perhitungan. Dalam praktiknya, untuk graf yang besar, Floyd's Algorithm lebih cepat daripada Warshall's Algorithm karena memiliki kompleksitas waktu yang lebih rendah, yaitu $O(n^3)$ dibandingkan dengan $O(n^3 \log n)$ pada Warshall's Algorithm. Oleh karena itu, Floyd's Algorithm lebih disukai dalam menyelesaikan perhitungan penutup transitif pada graf yang besar.

Namun, dalam kasus graf yang memiliki sifat-sifat tertentu, misalnya ketika graf tersebut bersifat sangat jarang (sparse), Warshall's Algorithm mungkin lebih cepat daripada Floyd's Algorithm karena memiliki kompleksitas waktu yang lebih rendah untuk graf yang bersifat sangat jarang. Dalam kesimpulannya, waktu yang dibutuhkan untuk menyelesaikan perhitungan penutup transitif dari sebuah graf sangat tergantung pada kompleksitas graf tersebut dan jenis algoritma yang digunakan. Dalam praktiknya, pilihan algoritma harus didasarkan pada sifat graf yang akan dihitung dan kompleksitas waktu yang dibutuhkan untuk menyelesaikan perhitungan.

E. REFERENSI

Introduction to the Design and Analysis of Algorithms (3rd ed.) [Levitin 2011-10-09]

An Introduction to the Analysis of Algorithms (2nd ed.) [Sedgewick & Flajolet 2013-01-28]

Misra, S., & Singh, R. (2020). Design and implementation of shortest path algorithm using Floyd Warshall algorithm. In 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 68-73). IEEE

<https://github.com/rehan909090/>