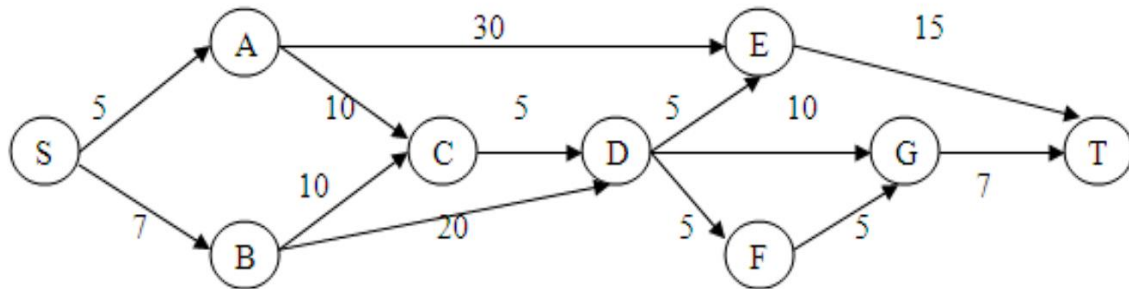


Nama : I Gede Arengga Nabakta

NIM : 1203220108

Kelas : IF 02-03

Mata Kuliah Kecerdasan Buatan



Berdasarkan gambar diatas tentukan Solusi terpendek dari S ke T

1. Semua kota di lewati ( dengan menganggap arah tidak ada )

Agar bisa melewati seluruh kota maka diperlukan metode BFS ( Breadth-First Search ) dengan Langkah Langkah sebagai berikut

- a. Berawal dari S dan pergi ke B dengan nilai 5
- b. Menganggap arah tidak ada maka setelah dari B maka akan pindah ke A jika melihat level atau tingkatan dari graph tersebut
- c. Lalu pindah ke C dan berlanjut lagi ke D.
- d. Jika terdapat percabangan maka kita akan memulai dari kanan. Dengan melihat graph diatas maka dari D akan berpindah ke F lalu dilanjutkan ke E.
- e. Setelah itu pindah ke level berikutnya atau G dan berakhir pada level paling bawah yaitu T

Sehingga urutan dari S menuju T adalah : **S - B - A - C - D - F - E - G - T**

2. Tidak melewati semua kota ( dengan menganggap arah ada )

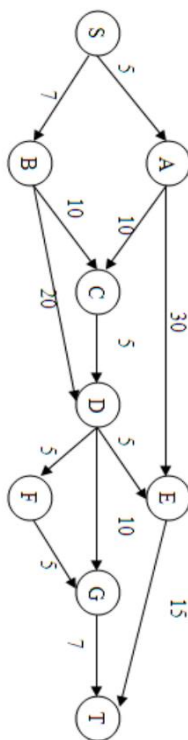
Agar bisa sampai di T dengan rute terpendek tanpa memikirkan cost atau value dari jarak/poin. Maka didapatkan rute terpenndek adalah **S - A - E - T**, tetapi jika memikirkan value maka kita bisa menggunakan Algoritma djikstra dengan Langkah Langkah sebagai berikut :

- a. Berawal di S kita lihat nilai terkecil antara A dan B. dan value yang terbaik adalah A dengan nilai 5. (**S - A**)
- b. Dari A kita melihat nilai dari C dan E. value C bisa di dapat dari A ataupun B. valu tersingkat dimiliki oleh A dengan total 15. Sedangkan nilai E adalah 30 sehingga kita pergi ke C. (**S - A - C**)
- c. Setelah dari C kita pergi ke D karena C tidak memiliki percabangan lainnya. (**S - A - C - D**)
- d. Dari D sendiri memiliki 3 percabangan yaitu E, F, dan G. value terkecil dimiliki oleh E dan F. sehingga mendapatkan hasil (**S - A - C - D - E**) atau (**S - A - C - D - F**).

- e. Setelah itu bisa dilihat E bisa langsung ke T dengan value 15. Sehingga didapatkan hasil **(S-A-C-D-E-T)** dengan total perolehan sebesar 40.
- f. Dilanjutkan untuk hasil kedua yaitu ke F lalu lanjut ke G dengan nilai 5. G sendiri bisa di raih dari F dengan nilai 5 atau dari D dengan nilai 10. Kedua hal tersebut memiliki nilai yang sama yaitu 10 sehingga mendapatkan hasil **(S-A-C-D-F-G)** atau **(S-A-C-D-G)**.
- g. Dari dengan kedua hasil itu kita bisa pergi ke F menjadikan hasil akhir **(S-A-C-D-F-G-T)** dengan value 37 dan kita bisa ke T tanpa ke F dengan rute **(S-A-C-D-G-T)** dengan nilai 37 juga. Sehingga daripada melalui **(S-A-C-D-E-T)** dengan total perolehan sebesar 40.

Jadi rute tercepat adalah melalui **(S-A-C-D-F-G-T)** ataupun **(S-A-C-D-G-T)**

### 3. BFS dan DFS



- a. BFS ( BREADTH-FIRST SEARCH ) adalah cara mencari rute dengan cara searching pada node yang memiliki level yang sama. Dalam hal level paling atas adalah S lalu disusul B dan A lalu C lalu D lalu F dan E lalu G lalu T sehingga jika menggunakan metode BFS didapatkan hasil sebagai berikut

				T
		D	D	D
	E	E	E	E
	C	C	C	C
B	B	B	B	B
A	A	A	A	A
S	S	S	S	S

Keterangan		
Sudah Dikunjungi	Baru dikunjungi	Queue Aktif

Begini penjelasannya :

1. Berawal dari S kita mengunjungi A dan B.
2. Lalu dari A kita mengunjungi C dan E.
3. Queue dilanjutkan ke B dengan mengunjungi D dan C.
4. Queue dilanjutkan ke C yang mengunjungi D saja.
5. Queue dilanjutkan ke E yang mengunjungi T.
6. Dari T didapatkan dari E. dan E didapatkan dari A dan A didapatkan dari S. Sehingga didapat hasil rute terpendek adalah **(S-A-E-T)**

- b. DFS ( DEPTH-FIRST SEARCH ) adalah metode yang memeriksa semua anak dibawahnya sebelum berpindah pada anak yang selevel. Begini penjelasannya :

1. Mulai dari node S. Pilih salah satu tetangga dari S yang belum dikunjungi. Mari kita pilih A.
2. Sekarang kita berada di A. Pilih tetangga A yang belum dikunjungi. Kita bisa memilih C atau E. Mari kita pilih C.
3. Sekarang kita berada di C. Pilih tetangga C yang belum dikunjungi. Kita bisa memilih D atau A. Namun, A sudah dikunjungi, jadi kita pilih D.
4. Sekarang kita berada di D. Pilih tetangga D yang belum dikunjungi. Kita bisa memilih G atau F. Mari kita pilih G.
5. Sekarang kita berada di G. Pilih tetangga G yang belum dikunjungi. Kita bisa memilih T. Kita telah mencapai T, sehingga rute yang ditemukan adalah **(S – A – C – D – G – T)**

				T
			G	G
		F	F	F
		E	E	E
	D	D	D	D
	C	C	C	C
B	B	B	B	B
A	A	A	A	A
S	S	S	S	S

Keterangan
Sudah Selesai
Stack yang POP
Stack yang di hold
Cabang ditemukan

Atau bisa seperti

1. Mulai dari simpul S.
2. Jelajahi cabang dari S ke A.
3. Dari A, jelajahi cabang ke E.
4. Dari E, langsung menuju T.
5. Karena T adalah tujuan, berhenti.

Sehingga mendapatkan hasil yang sama seperti BFS yaitu **(S – A – E – T)**

		T
	E	E
	C	C
A	A	A
B	B	B
S	S	S

Keterangan
Sudah Selesai
Stack yang POP
Stack yang di hold
Cabang ditemukan