

Pencarian

Kecerdasan Buatan

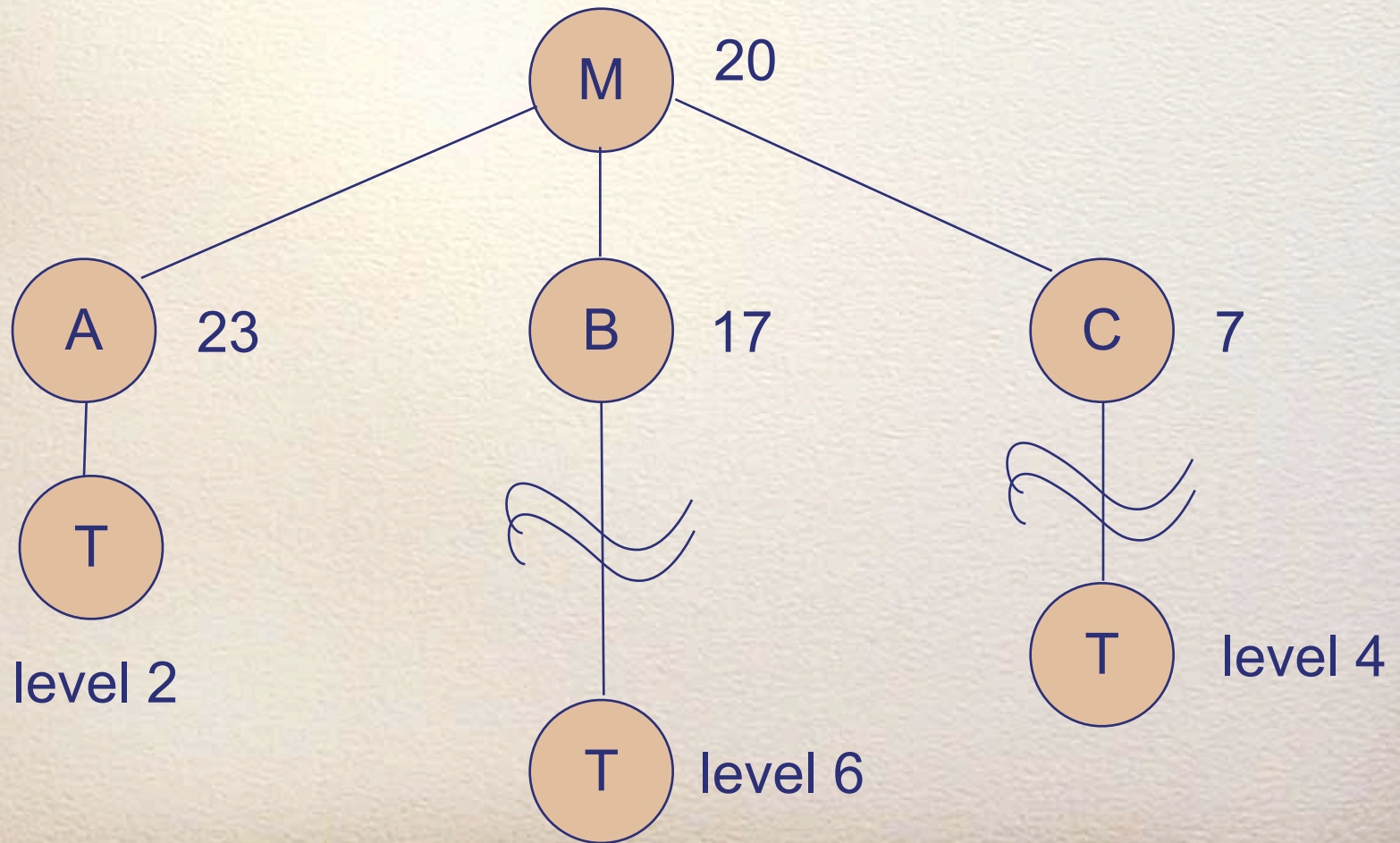
Pertemuan 4

Yudianto Sujana

Pencarian Heuristik

- Ada 4 metode pencarian heuristik
 - Pembangkit & Pengujian (Generate and Test)
 - Pendakian Bukit (Hill Climbing)
 - Pencarian Terbaik Pertama (Best First Search)
 - Simulated Annealing

Hill Climbing



Pencarian Terbaik Pertama (Best First Search)

- Metode best-first search ini merupakan kombinasi dari metode depth-first search dan metode breadth-first search dengan mengambil kelebihan dari kedua metode tersebut.
- Apabila pada pencarian dengan metode hill climbing tidak diperbolehkan untuk kembali ke node pada level yang lebih rendah meskipun node pada level yang lebih rendah tersebut memiliki nilai heuristik yang lebih baik, lain halnya dengan metode best-first search ini.
- Pada metode best-first search, pencarian diperbolehkan mengunjungi node yang ada di level yang lebih rendah, jika ternyata node pada level yang lebih tinggi ternyata memiliki nilai heuristik yang lebih buruk.

Pencarian Terbaik Pertama (Best First Search)

- Penentuan node berikutnya adalah node yang terbaik yang pernah dibangkitkan
- Menggunakan informasi
 - biaya perkiraan
 - biaya sebenarnya
- Terdapat 2 jenis
 - Greedy Best First Search
 - biaya perkiraan $f(n) = h(n)$
 - A^*
 - biaya perkiraan + biaya sebenarnya
 - $f(n) = g(n) + h(n)$

Pencarian Terbaik Pertama (Best First Search)

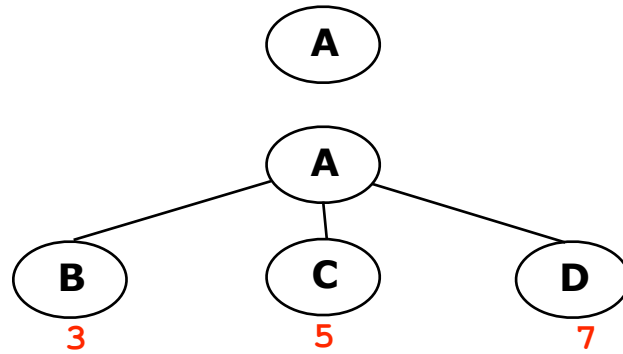
- Untuk mengimplementasikan metode ini menggunakan graph keadaan, dibutuhkan 2 antrian yang berisi node-node, yaitu:
 - OPEN, berisi node,node yang sudah dibangkitkan, namun belum diuji. Umumnya berupa antrian berprioritas yang berisi elemen-elemen dengan nilai heuristik tertinggi
 - CLOSED berisi node-node yang sudah diuji

Pencarian Terbaik Pertama (Best First Search)

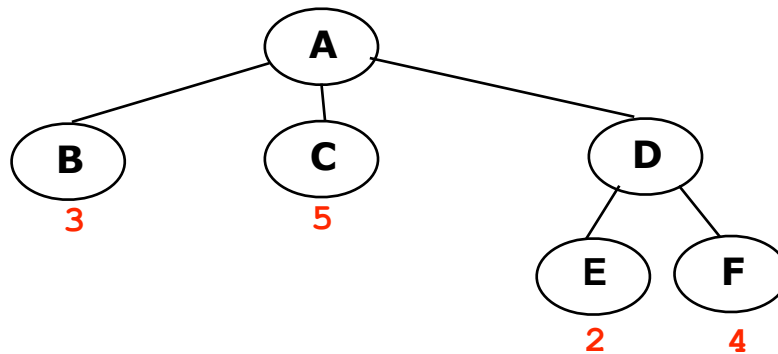
- Algoritma:
 - Tempatkan node awal A pada antrian OPEN.
 - Kerjakan langkah-langkah berikut hingga tujuan ditemukan atau antrian OPEN sudah kosong:
 - Ambil node terbaik dari OPEN;
 - Bangkitkan semua successornya;
 - Untuk tiap-tiap successor kerjakan:
 - Jika node tersebut belum pernah dibangkitkan sebelumnya, evaluasi node tersebut dan masukkan ke OPEN;
 - Jika node tersebut sudah pernah dibangkitkan sebelumnya, ubah parent jika lintasan baru lebih menjanjikan. Hapus node tersebut dari antrian OPEN.

Antrian OPEN

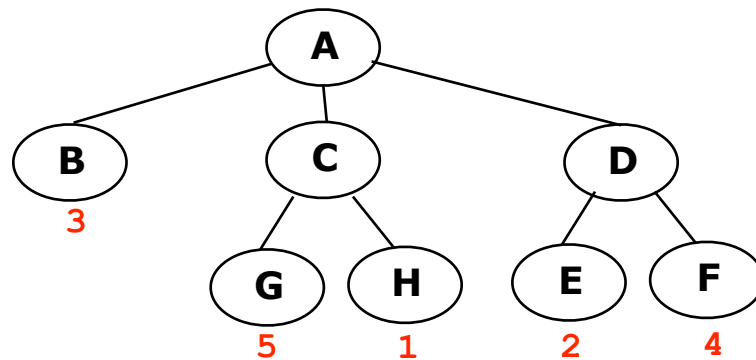
[A]



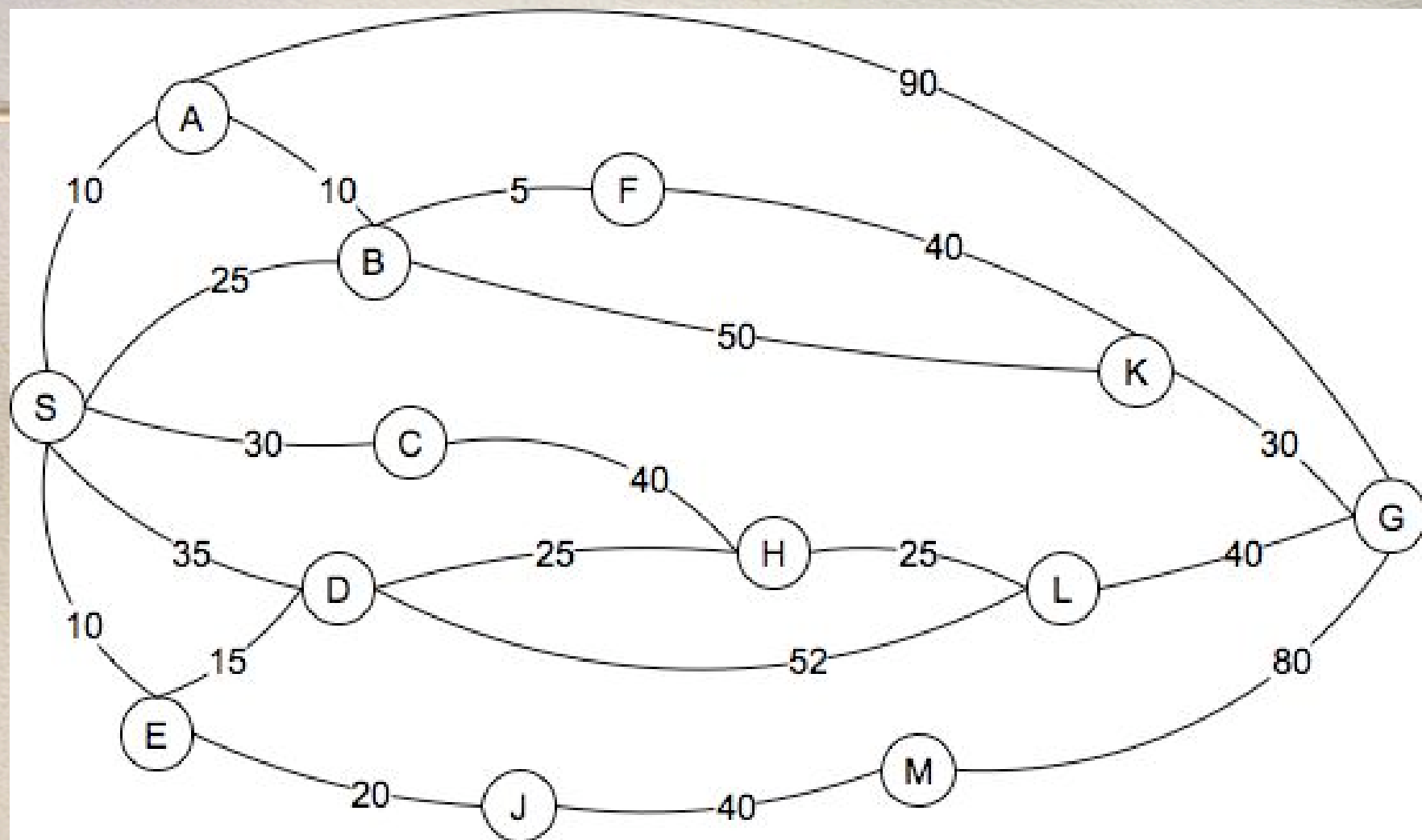
[D, C, B]



[C, F, B, E]

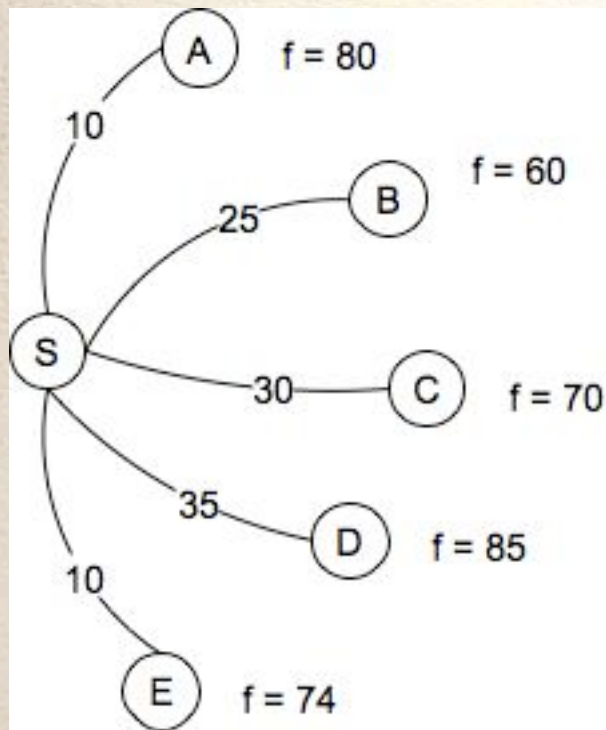


[G, F, B, E, H]



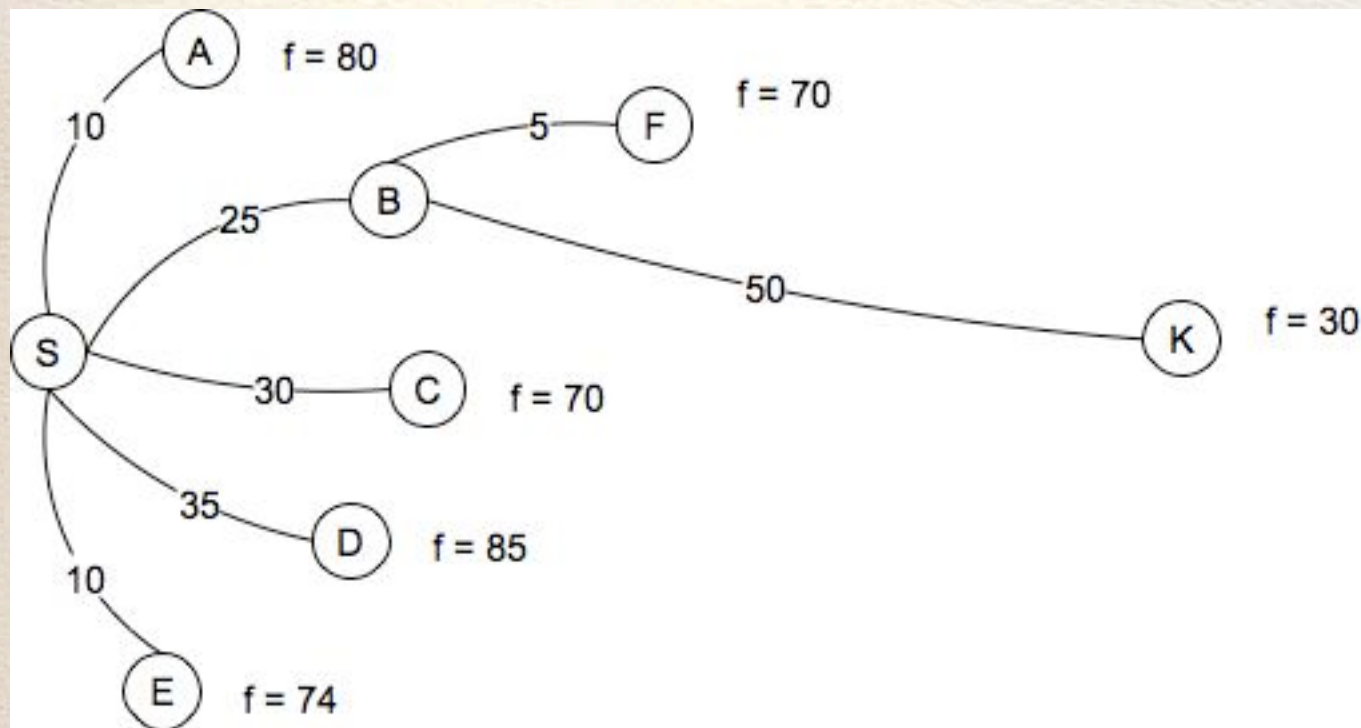
n	S	A	B	C	D	E	F	G	H	J	K	L	M
$h(n)$	80	80	60	70	85	74	70	0	40	100	30	20	70

Greedy Best First Search



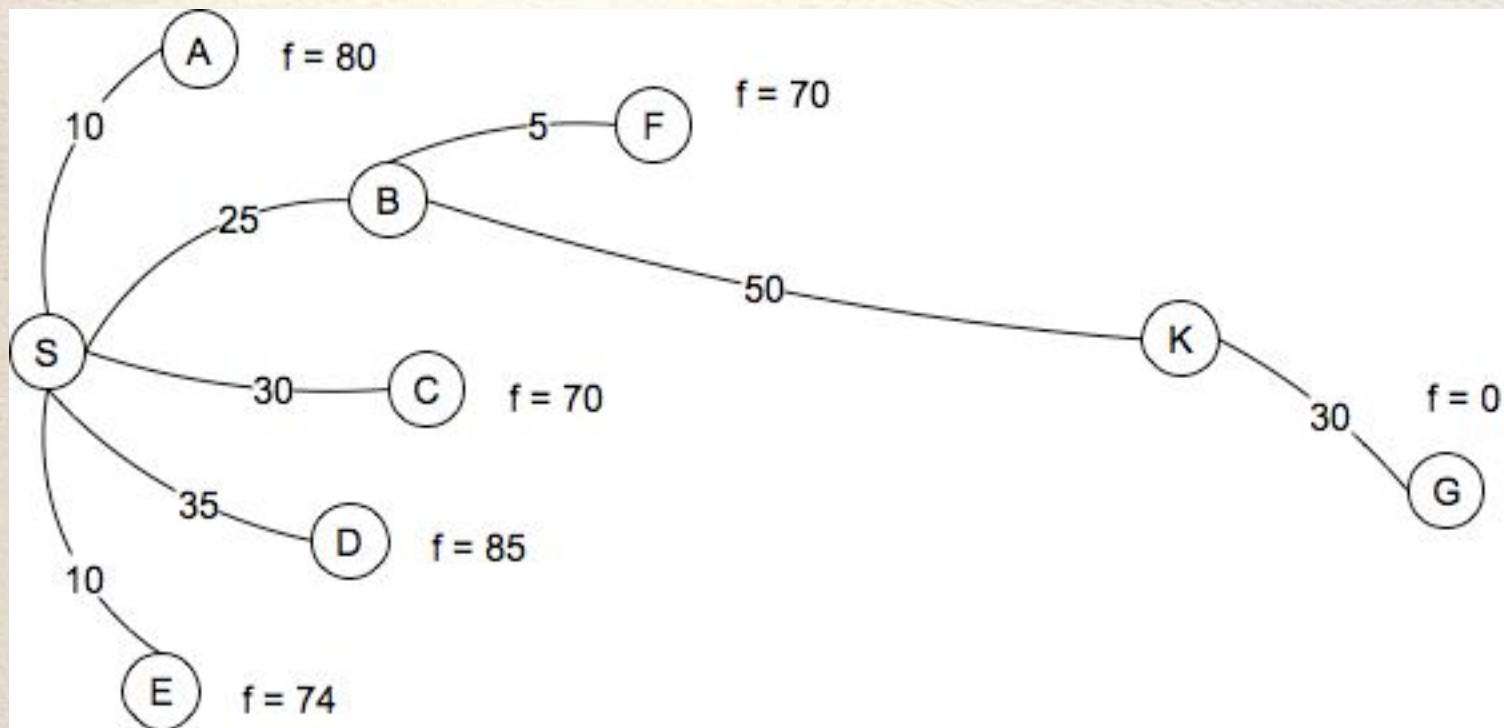
OPEN = [A,B,C,D,E]
CLOSED = [S]

Greedy Best First Search



OPEN = [A,C,D,E,F,K]
CLOSED = [S,B]

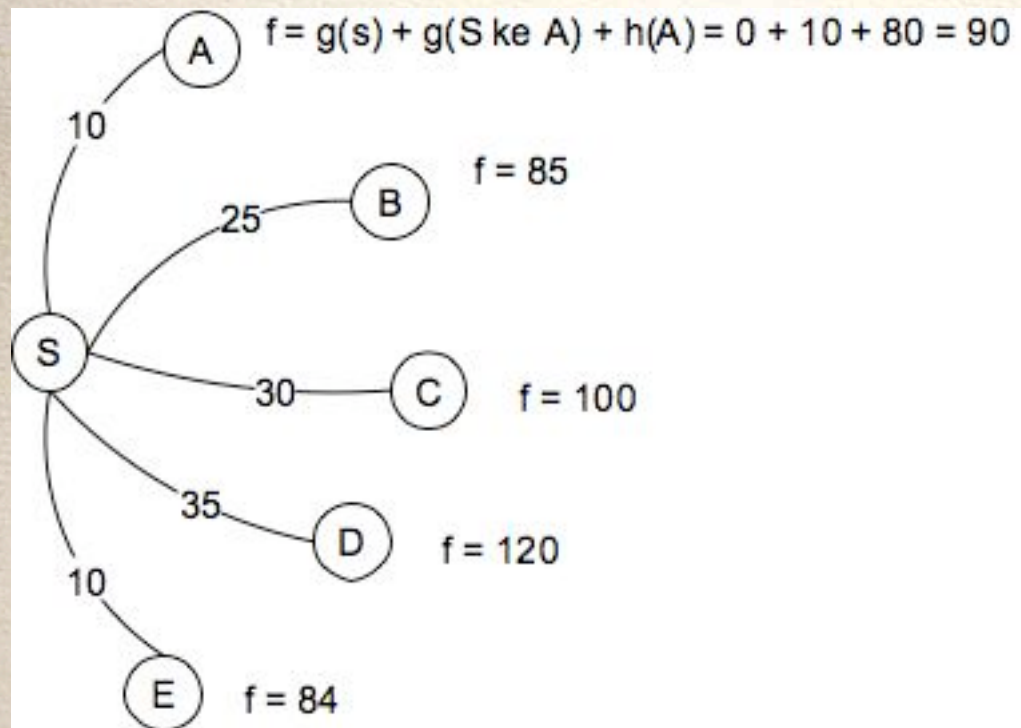
Greedy Best First Search



OPEN = [A,C,D,E,F,G]
CLOSED = [S,B,K]

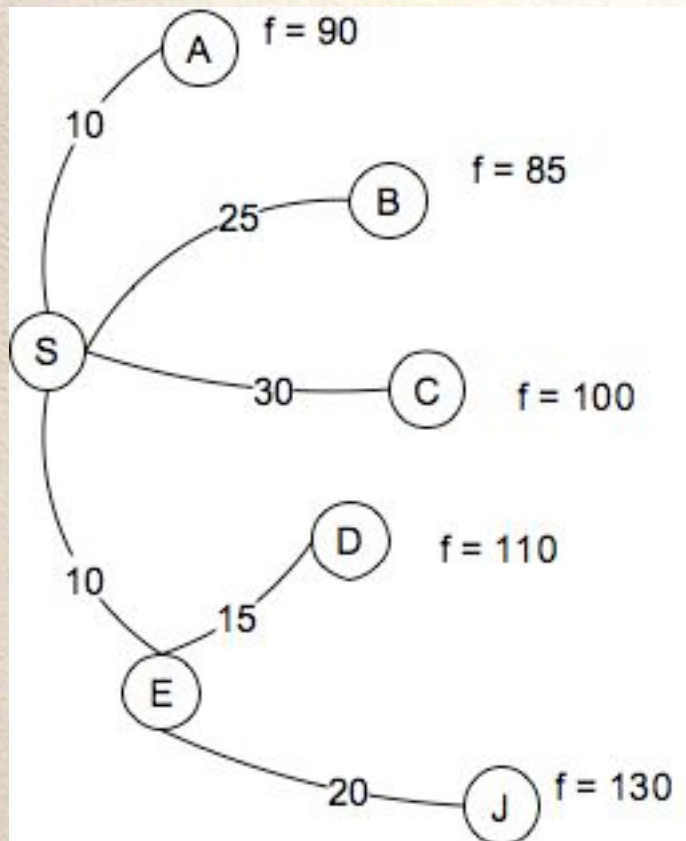
HASIL = S,B,K,G

A*



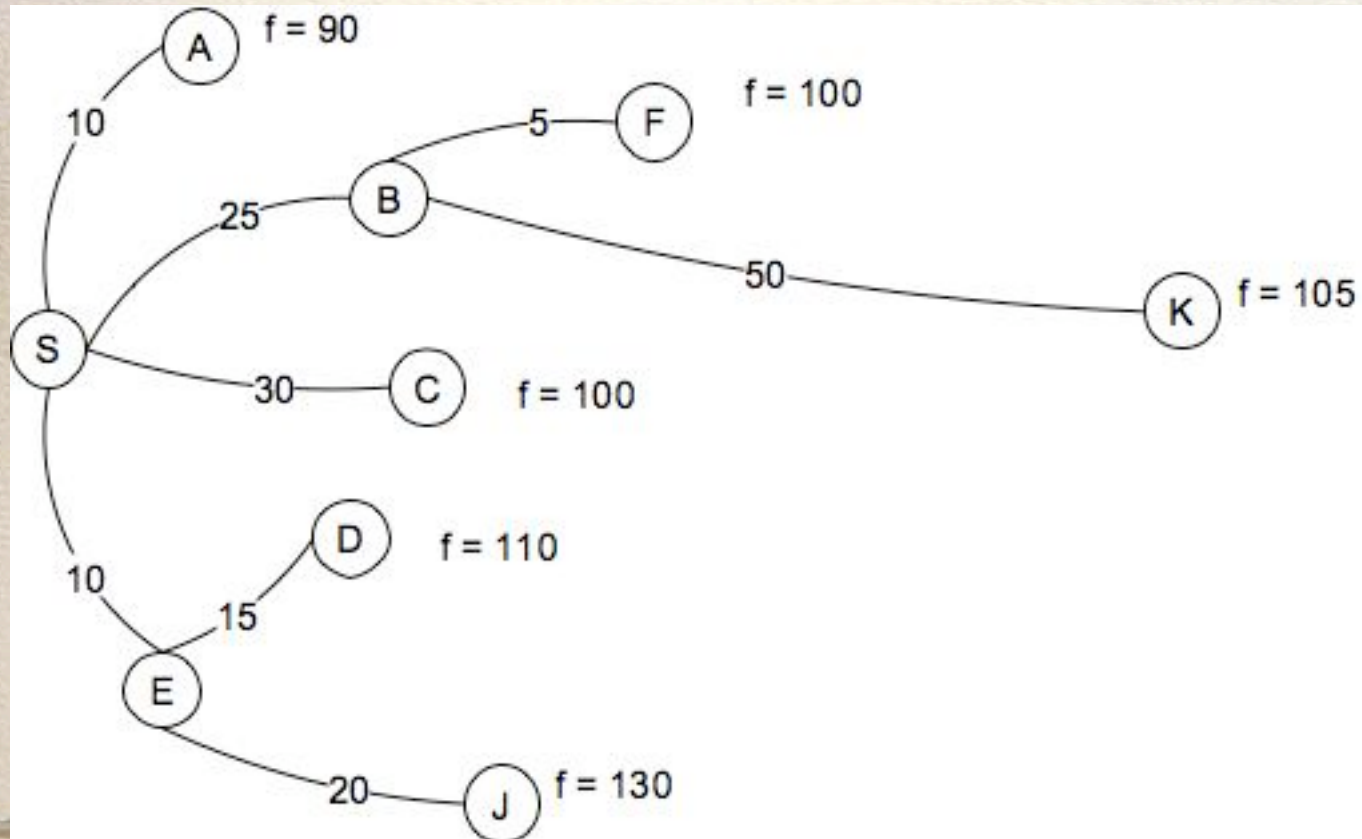
OPEN = [A,B,C,D,E]
CLOSED = [S]

A*



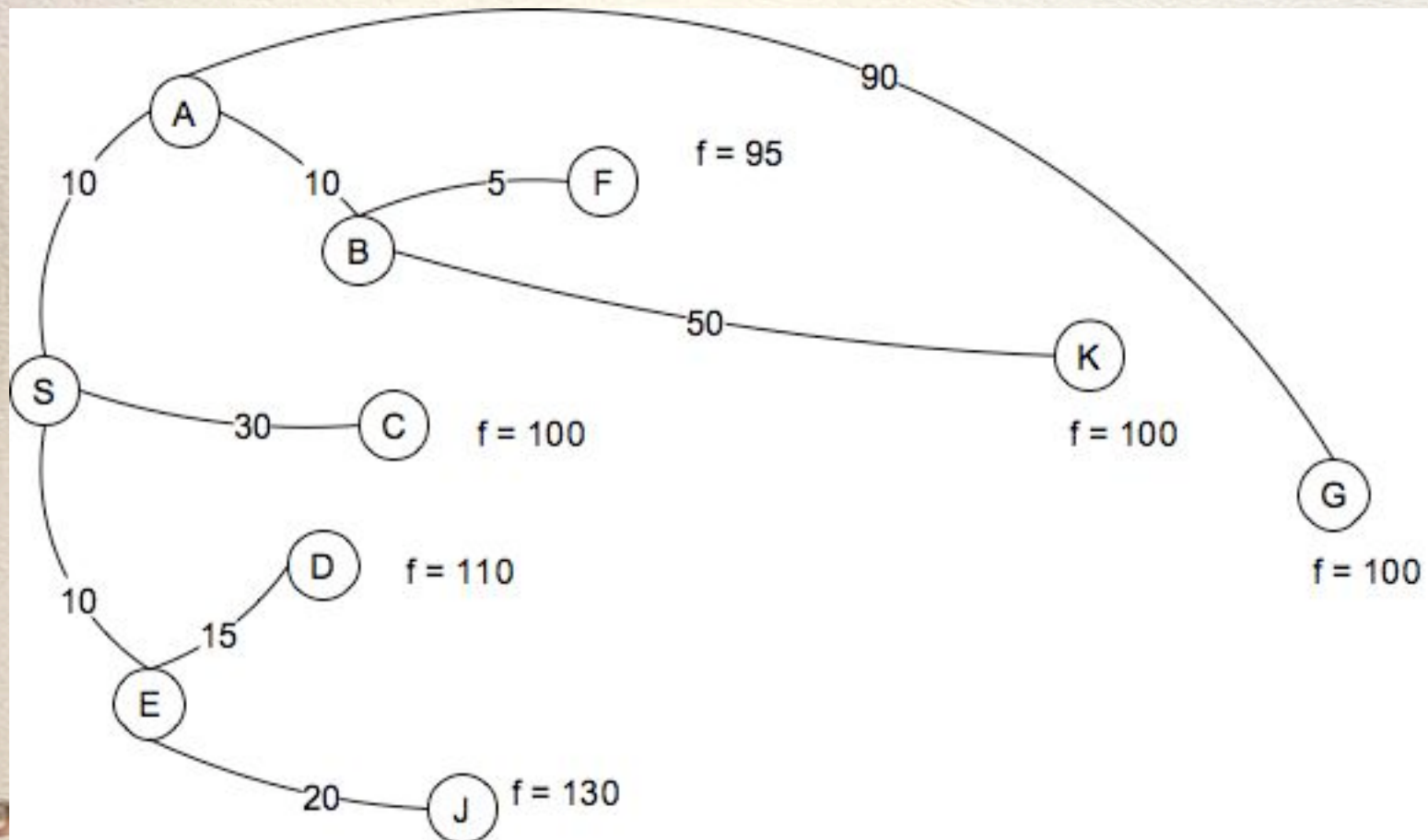
OPEN = [A,B,C,D,J]
CLOSED = [S,E]

A*



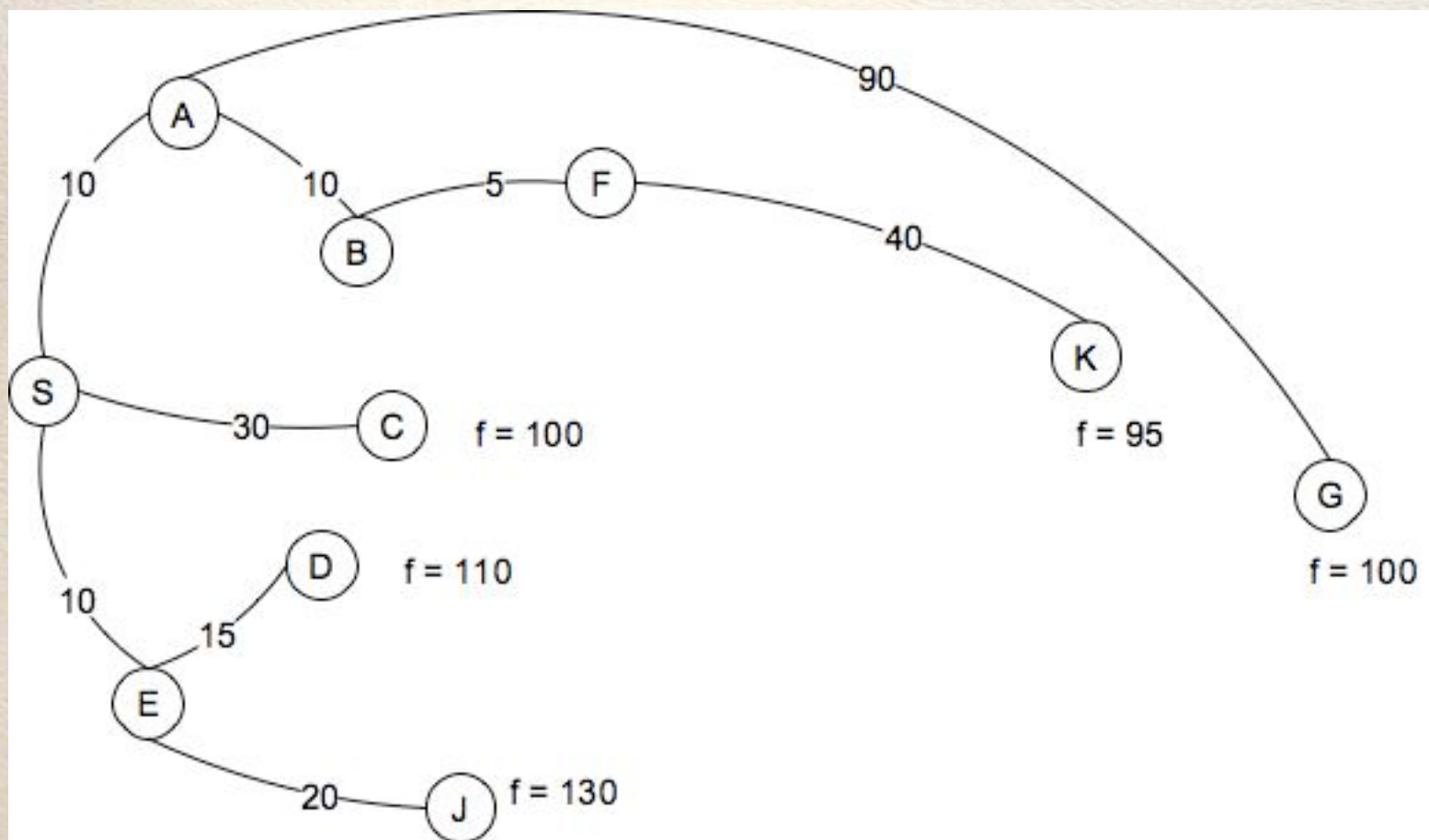
OPEN = [A,C,D,F,J,K]
CLOSED = [S,E,B]

A*



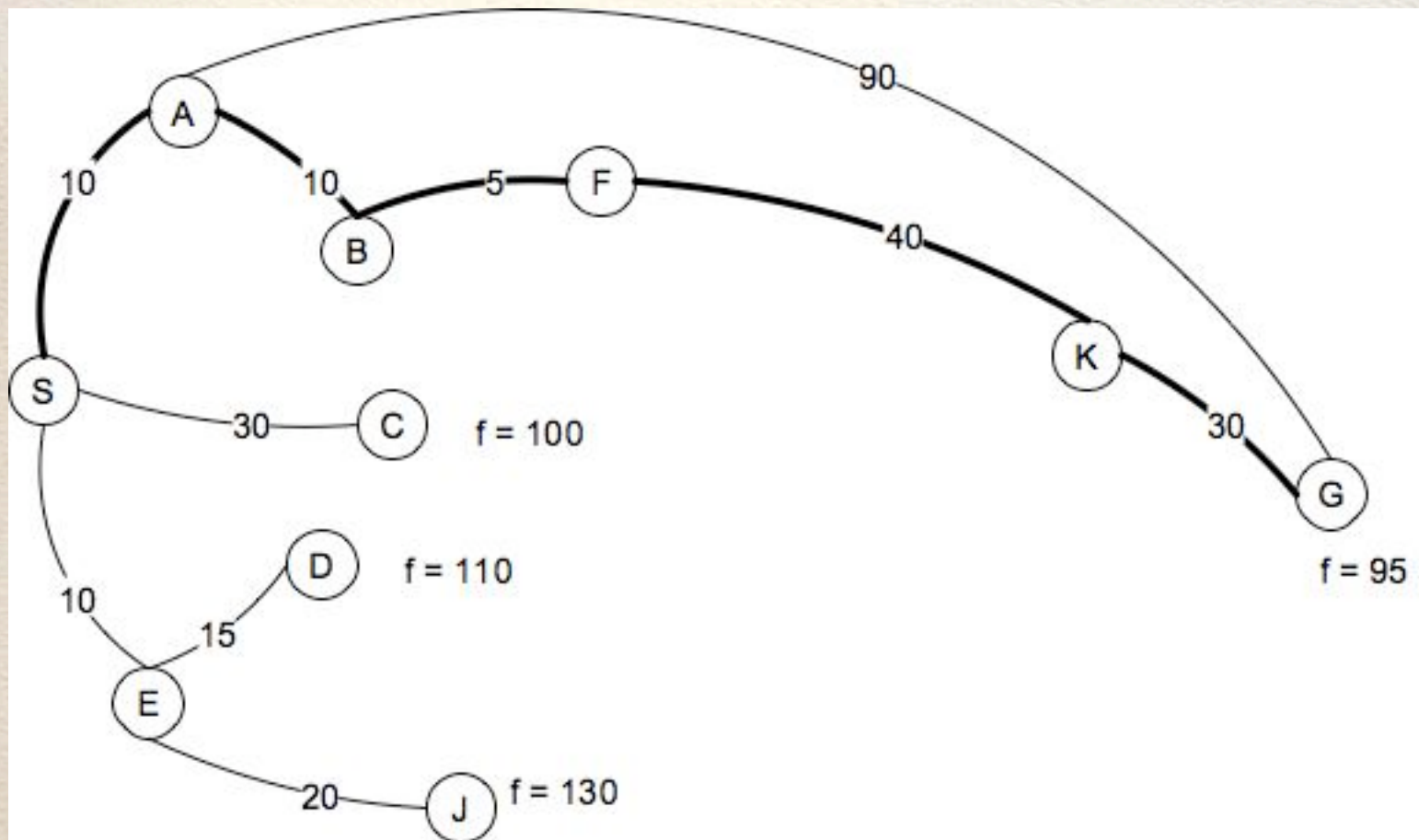
OPEN = [C,D,F,G,J,K]
CLOSED = [S,E,B,A]

A*



OPEN = [C,D,G,J,K]
CLOSED = [S,E,B,A,F]

A*



OPEN = [C,D,G,J]

CLOSED = [S,E,B,A,F,K]

HASIL = S,A,B,F,K,G

Any Questions?
