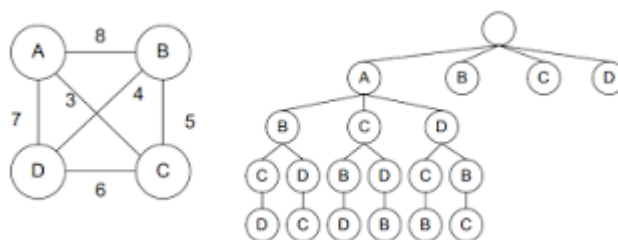


TEKNIK PENCARIAN HEURISTIK (HEURISTIC SEARCH)

- Heuristik adalah sebuah teknik yang mengembangkan efisiensi dalam proses pencarian, namun dengan kemungkinan mengorbankan kelengkapan (completeness).
- Fungsi heuristik digunakan untuk mengevaluasi keadaan-keadaan problema individual dan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diinginkan.
- Jenis-jenis Heuristic Searching:
 - Generate and Test.
 - Hill Climbing.
 - Best First Search.
 - Means-End Analysis, Constraint Satisfaction, dll.

1). PEMBANGKITAN dan PENGUJIAN (Generate and Test)

- Metode ini merupakan penggabungan antara depth-first search dengan pelacakan mundur (backtracking), yaitu bergerak ke belakang menuju pada suatu keadaan awal.
- **Algoritma :**
 1. Bangkitkan suatu kemungkinan solusi (membangkitkan suatu titik tertentu atau lintasan tertentu dari keadaan awal).
 2. Uji untuk melihat apakah node tersebut benar-benar merupakan solusinya dengan cara membandingkan node tersebut atau node akhir dari suatu lintasan yang dipilih dengan kumpulan tujuan yang diharapkan.
 3. Jika solusi ditemukan, keluar. Jika tidak, ulangi kembali langkah pertama.
- **Contoh :** “Travelling Salesman Problem (TSP)”
 - *) Seorang salesman ingin mengunjungi n kota. Jarak antara tiap-tiap kota sudah diketahui. Kita ingin mengetahui ruter terpendek dimana setiap kota hanya boleh dikunjungi tepat 1 kali. Misalkan ada 4 kota dengan jarak antara tiap-tiap kota seperti berikut ini :

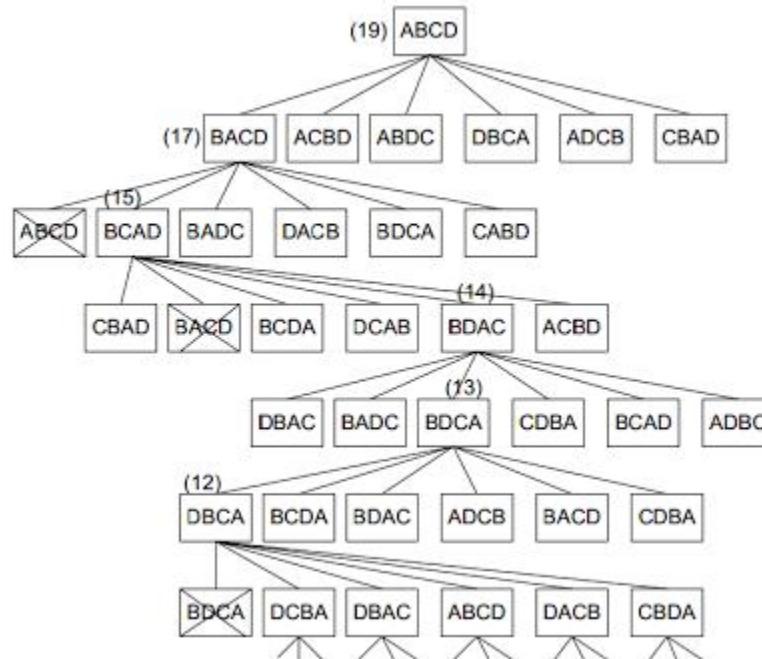


Alur pencarian dengan *Generate and Test*

Pencarian ke-	Lintasan	Panjang Lintasan	Lintasan terpilih	Panjang Lintasan terpilih
1	ABCD	19	ABCD	19
2	ABDC	18	ABDC	18
3	ACBD	12	ACBD	12
4	ACDB	13	ACBD	12
5	ADBC	16	ACBD	12
Dst.....				

2) PENDAKIAN BUKIT (Hill Climbing)

- Metode ini hampir sama dengan metode pembangkitan dan pengujian, hanya saja proses pengujian dilakukan dengan menggunakan fungsi heuristic. Pembangkitan keadaan berikutnya tergantung pada feedback dari prosedur pengetesan. Tes yang berupa fungsi heuristic ini akan menunjukkan seberapa baiknya nilai terkaan yang diambil terhadap keadaan-keadaan lain yang mungkin.
- Algoritma:**
 - Cari operator yang belum pernah digunakan; gunakan operator ini untuk mendapatkan keadaan yang baru.
 - Kerjakan langkah-langkah berikut sampai solusinya ditemukan atau sampai tidak ada operator baru yang akan diaplikasikan pada keadaan sekarang : Cari operator yang belum digunakan; gunakan operator ini untuk mendapatkan keadaan yang baru.
 - Evaluasi keadaan baru tersebut :
 - Jika keadaan baru merupakan tujuan, keluar
 - Jika bukan tujuan, namun nilainya lebih baik daripada keadaan sekarang, maka jadikan keadaan baru tersebut menjadi keadaan sekarang.
 - Jika keadaan baru tidak lebih baik daripada keadaan sekarang, maka lanjutkan iterasi.
- Contoh:** TSP dengan Simple Hill Climbing Disini ruang keadaan berisi semua kemungkinan lintasan yang mungkin. Operator digunakan untuk menukar posisi kota-kota yang bersebelahan. Apabila ada n kota, dan kita ingin mencari kombinasi lintasan dengan menukar posisi urutan 2 kota, maka kita akan mendapatkan sebanyak $n!/2!(n-2)!$ atau sebanyak 6 kombinasi. Fungsi heuristic yang digunakan adalah panjang lintasan yang terjadi.

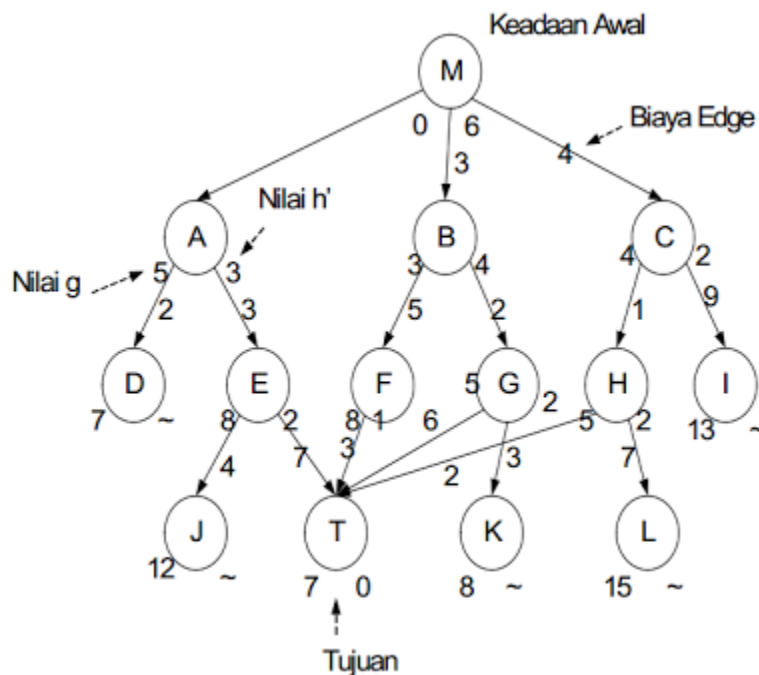


3). PENCARIAN TERBAIK PERTAMA (Best-First Search)

- Metode ini merupakan kombinasi dari metode depthfirst search dan breadth-first search. 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 heuristic yang lebih buruk.
- Fungsi Heuristik yang digunakan merupakan prakiraan (estimasi) cost dari initial state ke goal state, yang dinyatakan dengan :

$$f^*(n) = g(n) + h'(n)$$

- f^* = Fungsi evaluasi
- g = cost dari initial state ke current state
- h' = prakiraan cost dari current state ke goal state
- Contoh:** Misalkan kita memiliki ruang pencarian seperti pada gambar berikut. Node M merupakan keadaan awal dan node T merupakan tujuannya. Biaya edge yang menghubungkan node M dengan node A adalah biaya yang dikeluarkan untuk bergerak dari kota M ke kota A. Nilai g diperoleh berdasarkan biaya edge minimal. Sedangkan nilai h' di node A merupakan hasil perkiraan terhadap biaya yang diperlukan dari node A untuk sampai ke tujuan. $h'(n)$ bernilai \sim jika sudah jelas tidak ada hubungan antara node n dengan node tujuan (jalan buntu). Kita bisa merurut nilai untuk setiap node.



Tabel status tiap node

Node (n)	$g(n)$	$h'(n)$	$f(n)$
M	0	6	6
A	5	3	8
B	3	4	7
C	4	2	6
D	7	~	~
E	8	2	10
F	8	1	9
G	5	2	7
H	5	2	7
I	13	~	~
J	12	~	~
K	8	~	~
L	15	~	~
T	7	0	7

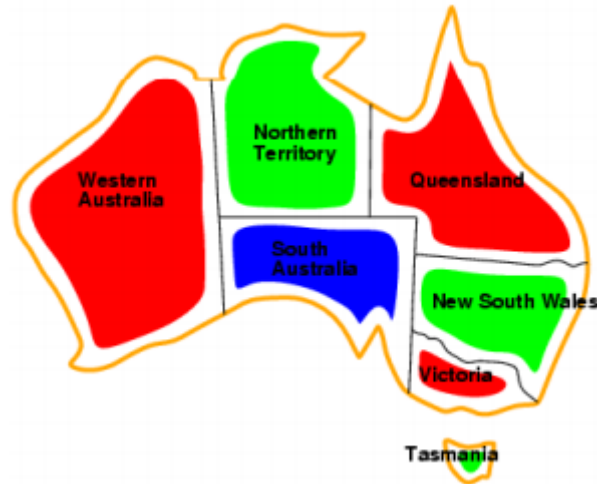
Solusi : M-C H-T

4).

Constraint

Satisfaction

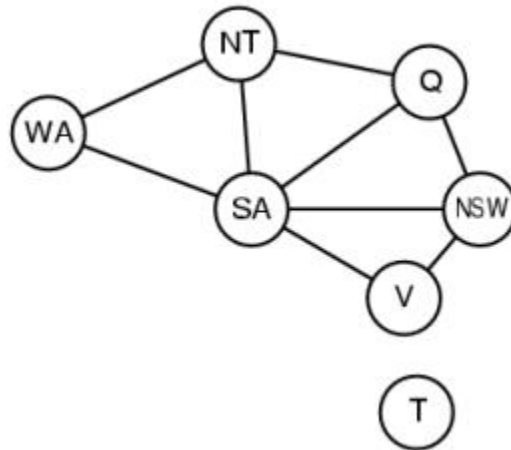
- Problem search standard :
 - state adalah "black box"
 - setiap struktur data yang mendukung fungsi successor, fungsi heuristik dan tes goal.
- CSP:
 - state didefinisikan sebagai variabel X_i dengan nilai dari domain D_i
 - Tes goal adalah sekumpulan constraint yang menspesifikasikan kombinasi dari nilai subset variabel.
- Contoh sederhana adalah bahasa representasi formal.
- CSP ini merupakan algoritma general-purpose dengan kekuatan lebih daripada algoritma pencarian standar.
- **Contoh** : Pewarnaan Peta



- Variabel WA, NT, Q, NSW, V, SA, T
- Domain $D_i = \{\text{red, green, blue}\}$
- Constraints : daerah yang bertetangga dekat harus memiliki warna yang berbeda.
- Contoh $WA \neq NT$, atau $(WA, NT) \{(\text{red, green}), (\text{red, blue}), (\text{green, red}), (\text{green, blue}), (\text{blue, red}), (\text{blue, green})\}$
- Solusi lengkap dan konsisten, contoh : $WA = \text{red}, NT = \text{green}, Q = \text{red}, NSW = \text{green}, V = \text{red}, SA = \text{blue}, T = \text{green}$

Constraint Graf

- Binary CSP biner : setiap constraint merelasikan dua variabel
- Graf Constraint : node adalah variabel, arc adalah constraint



5). MEA (Means-Ends Analysis)

- MEA adalah strategi penyelesaian masalah yang diperkenalkan pertama kali dalam GPS (General Problem Solver) [Newell & Simon, 1963].
- Proses pencarian berdasarkan ruang masalah yang menggabungkan aspek penalaran forward dan backward.
- Perbedaan antara state current dan goal digunakan untuk mengusulkan operator yang mengurangi perbedaan itu.

- Keterhubungan antara operator dan perbedaan tsb disajikan sebagai pengetahuan dalam sistem (pada GPS dikenal dengan Table of Connections) atau mungkin ditentukan sampai beberapa pemeriksaan operator jika tindakan operator dapat dipenetrasi.
- Contoh OPERATOR first-order predicate calculus dan operator2 tertentu mengijinkan perbedaan korelasi task-independent terhadap operator yang mengurangnya.
- Kapan pengetahuan ada tersedia mengenai pentingnya perbedaan, perbedaan yang paling utama terpilih pertama lebih lanjut meningkatkan rata-rata capaian dari MEA di atas strategi pencarian Brute-Force.
- Bagaimanapun, bahkan tanpa pemesanan dari perbedaan menurut arti penting, MEA meningkatkan metode pencarian heuristik lain (di rata-rata kasus) dengan pemusatan pemecahan masalah pada perbedaan yang nyata antara current state dengan goal-nya.