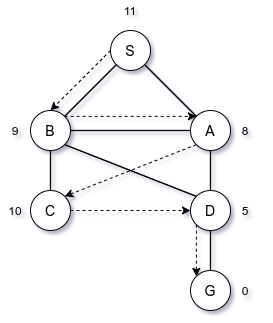
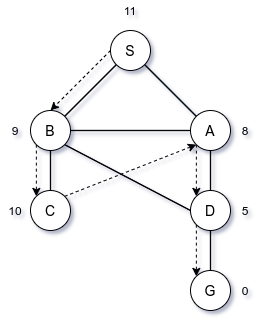
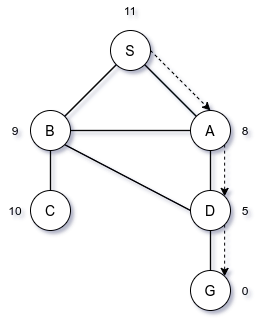
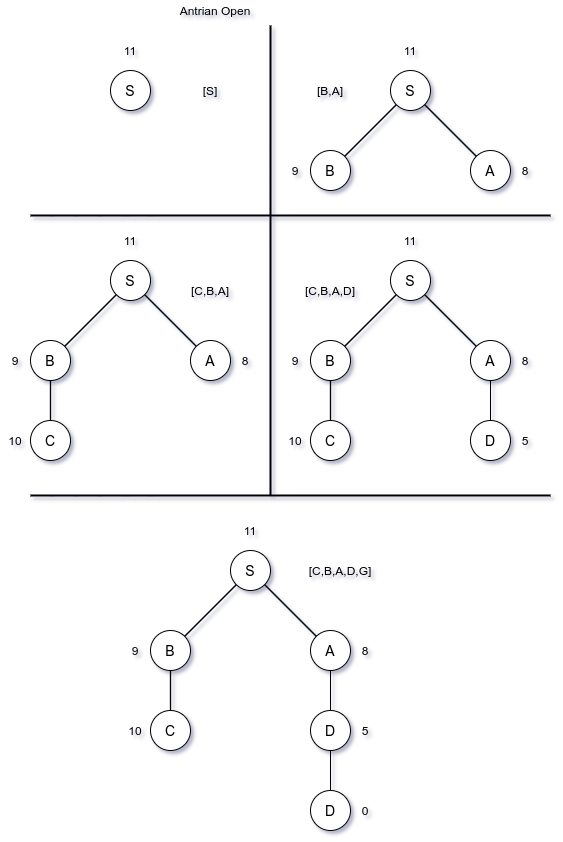
UTS Artificial Intelegence

Nama : M Faisal Halim Mata Kuliah : Artificial Intelegence

Nim : 19.240.0163 Ruang : -

Kelas : 4P43 Tanggal : May 5, 2021

Progdi : Teknik Informatika Dosen : Taryadi, M.Cs

1. AI adalah teknologi yang mensimulasikan kcerdasan manusia kedalam mesih (komputer) untuk menyelesaikan berbagai persoalan sebaik atau bahkan lebih baik dari manusia.  
     
   Penerapan AI pada aplikasi komersial, sepert Google Assistan, SIRI. contoh lain dalam aplikasi POST (Point Of Sale). pada aplikasi AI digunakan untuk mengotomatisasikan pengitungan keuangan dan stok barang. serta pembuatan grafik untuk membuat informasi lebih menarik.
2. - Kecerdasan buatan lebih bersifat permanen, karena semua data yang diperlukan disimpan dalam penyimpanan yang terbilang teknologi yang maju.  
     
   - Sementara kecerdasan alamai akan cenderung berubah dalam kurun waktu tertentu atau berdasarkan faktor lain.
3. AI bisa melakukan apapun, seperti menghitung, membuat, membaca dll. tergantung dari bidang yang dibuatnya.  
     
   Program Konvesional biasanya hanya bisa melakukan 1 hal saja, contoh, calculator padaumumnya hanya bisa menghitung saja, tidak seperti membuat laporan / history. namun sekarang sudah berbeda.
4. - Mendefinisikan suatu ruang keadaan (state space)   
   - Menetapkan satu atau lebih keadaan awal (initial state)   
   - Menetapkan satu atau lebih tujuan (goal state)   
   - Menetapkan kumpulan aturan
5. -
6. Knowledge Base : adalaah fakta tentang object dalam domain yang berhubungan dengan domain-domain tersebut.  
     
   Interface Engine : kumpulan prosedur yang digunakan dalam menjawab pertanyaan, menyelesaikan masalah, atau membuat keputusa, semua itu bertujuan untuk menguji basis pengetauan.
7. - Completeness : apakah menjamin menemukan solusi jika emang ada.  
   - Optimality : apakah menjamin menemukan solusi terbaik dari beberapa solusi yang ada.  
   - Time dan Space Complexity : waktu dan memori yang dibutuhkan.
8. - BFS menggunakan antrian struktur data, DFS cenderung menggunakan data stack atau rekursif  
   - BFS Lebih optiman dari DFS, karena memang DFS belum optimal.  
   - DFS lebih effisien dalam menggunalan space memory daripada DFS.
9. - Hill Climbing : pencarian heuristik untuk masalah optimasi. pada hill climbing tidak diperbolehkan kembali ke node awal walaupun node tersebut memiliki nilai yang lebih baik.  
     
   - Best first search : tidak seperti hill climbing, pada bagian ini diperbolehkan kembali ke node awal jika node awal memiliki nilai yang lebih baik, atau node dengan level tinggi tidak sebaik node awal.
10.  **Breadth-firdt search**Semua node level n akan dikunjungi dahulu sebelum node pada level n+1, pencarian dimulai dari node root, lalu ke level 1, dan level selanjutnya sampaikahirnya solusi ditemukan  
      
    Jalur : S-B-A-C-D-G  
      
      
      
      
      
      
      
      
    **Depth-first search**Pencarian dilakukan dari node anak-anaknya sebelum dilakukan pencarian ke node-node yang selevel.  
      
    Jalur : S-B-C-A-D-G  
      
      
      
      
      
      
      
      
      
      
      
      
      
      
      
      
    **Hill climbing (steepest-ascent hill climbing)**Steepest-ascent hill climbing hampir sama dengan simple-ascent hill climbing, hanya saja gerakan pencarian tidak dimulai dari kiri, tetapi berdasarkan nilai heurisik terbaik.  
      
    Jalur : S-A-D-G  
      
      
      
      
      
      
      
    **Best first search**  
    Metode best first search merupakan kombinasi metode depth search dan breadth first search dengan mengambil kelebihan dari kedua metode tersebut. Hill climbing tidak diperbolehkan kembali ke node yang lebih rendah meskipun node tersebut nilai heuristik-nya lebih baik. Pada best first seearch, pencari diperbolehkan mengunjungi node yang lebih rendah, jika node pada level yang lebih tinggi memiliki nilai heuristik lebih buruk dari node level sebelumnya. Untuk mengimplementasikan metode ini, dibutuhkan 2 antrian yang berisi node-node, yaitu  
      
      
      
    Open: berisi node-node yang sudah dibangkitkan, memiliki fungsi heuristik namun belum diuji. Umumnya berupa antian berprioritas yang berisi elemen-elemen dengan nilai heuristik tertinggi.   
    Closed: berisi node-node yang sudah diuji.