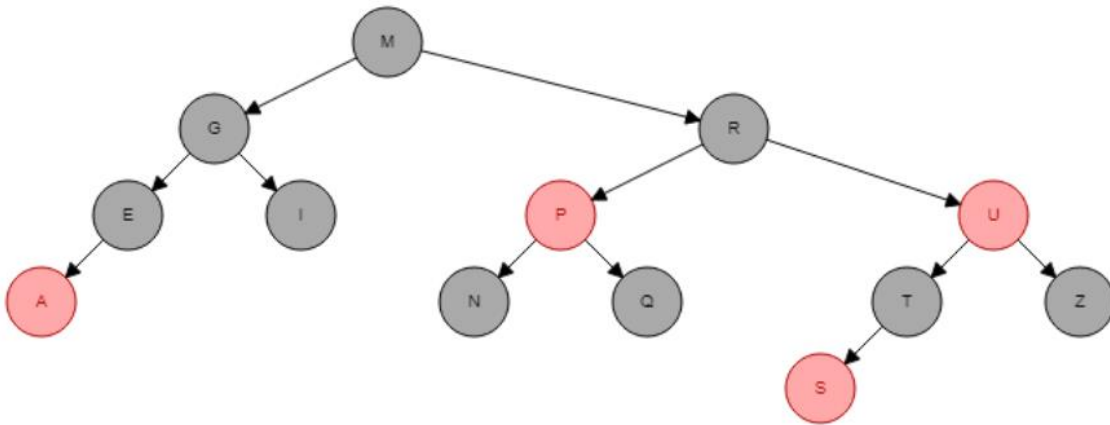


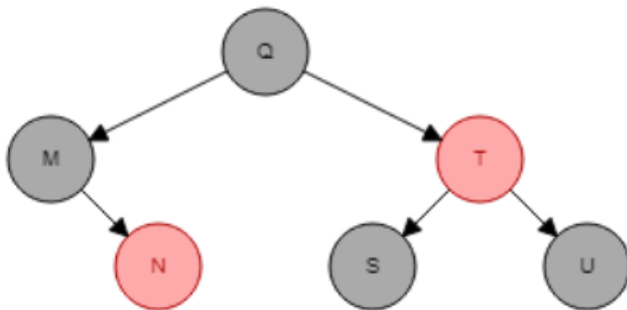
NAMA : RICO TANDRIO

Final Exam Bootcamp Even

Nomor 1a



Nomor 1b



Nomor 2

List	Weight	Visited
CD	10	V
DJ	10	V
CI	10	V
IH	12	V
HG	14	V
GA	14	V
JI	15	X
EH	15	V
FC	15	V
AB	16	V
EA	17	X
FD	20	X
FJ	20	X
BC	21	X
EB	23	X
FI	26	X
HC	28	X
EG	29	X
HB	30	X

TOTAL WEIGHT
116

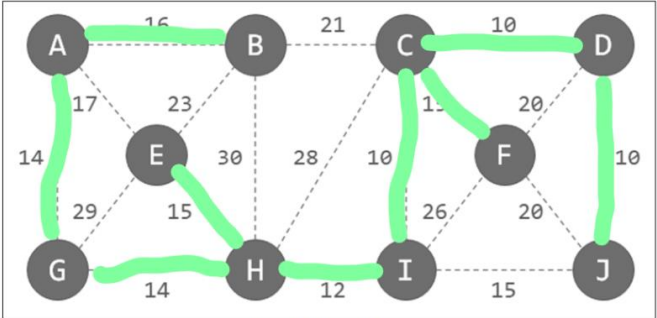


Figure 3 Undirected Graph

Nomor 3

a. **“Red Black Tree will not have any kind of 3-nodes chain”**

Pernyataan tersebut benar karena jika ada sebuah tree dengan 3 nodes yang berurutan, tree tersebut tidak mungkin bisa disebut sebagai red black tree karena tidak memenuhi atau melanggar aturan – aturan dari red black tree.

b. **“Every path from a node (including root) to any of its leaves (NULL) has the same number of black nodes which means the leaves always at the same height”**

Pernyataan tersebut tidak sepenuhnya salah karena merupakan salah satu dari 4 aturan untuk membentuk Red Black Tree, yaitu setiap path dari root ke null pointer N harus melewati jumlah node hitam yang sama yang nantinya path ini akan disebut sebagai unsuccessful search path karena path ini lah yang kita lalui untuk mencari sebuah node yang tidak terdapat dalam tree dengan N dianggap sebagai node berwarna hitam. *Hanya saja untuk jumlah leaves tidak ada aturan yang mengatakan bahwa akan selalu sama dengan height.*

c. **“Height (h) of the longest path in red black tree can have a maximum height of two times plus one ($2h+1$) from the shortest path”**

Pernyataan tersebut salah karena red black tree merupakan self-balancing binary search tree sama seperti AVL tree yang berarti balance factor dari red black tree tidak akan lebih dari 1 atau path terpanjang dari red black tree maksimalnya hanya path terpendek + 1.

d. **“The path from the root to the descendant nearest leaf has at least one black node and one red node”**

Pernyataan ini salah, menurut aturan dari red black tree setiap jalur dari root ke null pointer harus melewati node hitam. Di aturan ini tidak mengatakan bahwa path harus terdapat node berwarna merah.

Nomor 4

- a. **“The Red Black Trees are more balanced compared to AVL Trees”**

Pernyataan ini salah karena AVL Tree lebih balanced karena proses insert dan delete dari AVL relatif strict balancing.

- b. **“AVL Tree is suitable for applications that require a lot of insert and delete processes compare to Red Black Tree”**

Pernyataan tersebut salah karena proses insertion dan deletion pada AVL Tree lebih kompleks karena setiap insert dan delete akan dilakukan proses rotation berbeda dengan Redd Black Tree yang tidak selalu melakukan rotation disetiap insert dan delete.

- c. **“Inserting a set of number in AVL Tree and Red Black Tree will not produce the same tree shape”**

Pernyataan tersebut benar, karena walaupun AVL Tree dan Red Black Tree adalah self-balancing binary search tree tetapi metode yang digunakan untuk melakukan self-balancing berbeda. Di AVL Tree self-balancing dilakukan setiap melakukan insertion dan deletion, sedangkan di Red Black Tree setiap insertion dan deletion selalu memverifikasi aturan – aturan dari red black tree.

- d. **“Red Black Tree has slower searching process than AVL Tree”**

Pernyataan tersebut benar, karena walaupun Time Complexity untuk searching dari Red Black Tree dan AVL Tree dalam big O notation sama – sama $O(\log n)$. Namun karena AVL Tree relatif strict balancing maka efisiensi searching dari AVL Tree sedikit lebih baik dari Red Black Tree.