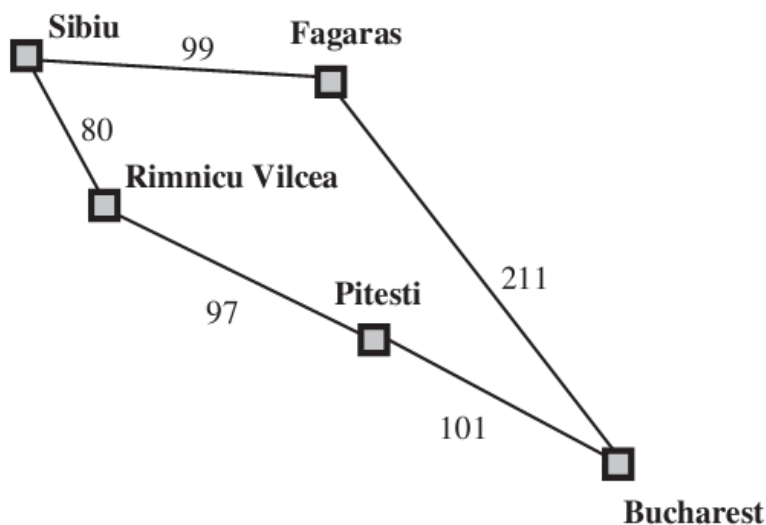
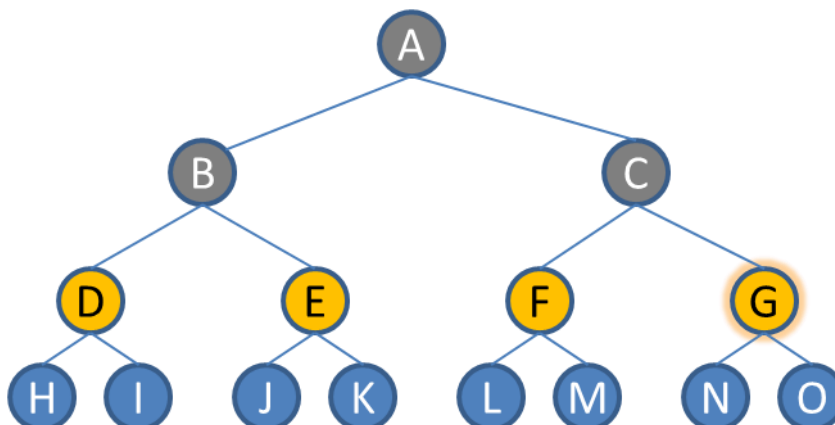


BUSCA CEGA – EXERCÍCIOS

1. Por que a estratégia de busca em largura só garante encontrar a **solução ótima** quando o custo por ação é uniforme?
2. Compare busca em largura (extensão) com custo uniforme para o espaço de estados apresentado na figura em termos de: nós criados (complexidade temporal), nós na memória (complexidade espacial), otimalidade (solução de menor custo) e completude (se consegue encontrar a solução). Utilizar busca em grafo sendo o estado inicial = Sibiu e o final = Bucharest.



3. Explique e exemplifique a diferença de **complexidade espacial**  $O(b^d)$  da busca em largura e da complexidade espacial  $O(b.m)$  da busca em profundidade. Exemplifique tomando por base o espaço de estados abaixo sendo G o estado objetivo e o custo das ações uniforme e igual a 1. Utilizar busca em grafo.



4. No espaço de estados acima, quantos nós devem ser gerados para se encontrar G na estratégia de busca em profundidade. Compare o resultado obtido com a complexidade espacial (quantos nós em memória).

5. Explique porque na busca em aprofundamento iterativo ainda se tem uma complexidade espacial e temporal similar a da busca em profundidade sendo que a busca é reiniciada a cada incremento de  $l$  (nível) a ser explorado.

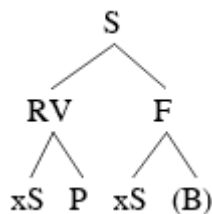
### RESPOSTA Q1.

Suponha uma situação onde os custos não são uniformes: há dois nós objetivos na árvore de busca  $a$  e  $b$ , sendo que a profundidade de  $a$  é 1 e o custo é 10 e a profundidade de  $b$ , 2 e o custo 7. A busca em largura achará antes o nó  $a$  e o retornará como solução – sendo que o caminho até  $a$  não é a solução ótima.

Se os custos forem uniformes, a solução ótima sempre será a mais rasa porque terá o menor número de AÇÕES e, portanto, menor custo.

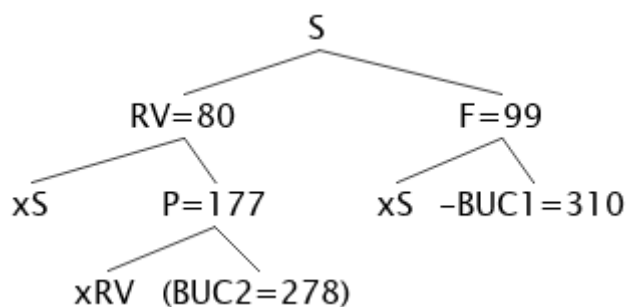
### RESPOSTA Q2.

**x<Nó>**: nó não incluído na fronteira porque **<Nó>.estado** já foi explorado  
**-<Nó>**: nó não incluído na fronteira porque **<Nó>.estado** já está na fronteira e é melhor ou nó substituído na fronteira porque **<Nó>.estado** gerado é melhor  
**(<Nó>)**: nó que representa a solução



- Complex. Espacial: **5** = todos os nós devem ser mantidos em memória.
- Complex. Temporal: **7** (considerando que S foi gerado 3 vezes). Outra contagem possível seria 5 se desconsiderarmos a geração dos nós 'xS'
- Completo? **Sim, como b é finito consegue encontrar a solução sempre que existir**
- Ótimo? **Não**, encontra o nó objetivo mais raso cujo custo é 310 (mas não é o caminho mais curto que passa por PITESTI)

### BUSCA CUSTO UNIFORME



[S [RV=80 [xS][P=177 [xRV][(BUC2=278)]]][F=99 [xS][-BUC1=310]]]

- Complex. Espacial: **5** (observe que BUC2 substitui BUC2)
- Complex. Temporal: **6** (foram gerados 6 nós até alcançar o nó objetivo). 9 se contar os nós que já haviam sido explorados (precedidos por x)
- Completo? **Sim, como b é finito consegue encontrar a solução sempre que existir**
- Ótimo? **Sim**, encontra o nó objetivo que está no caminho mais curto

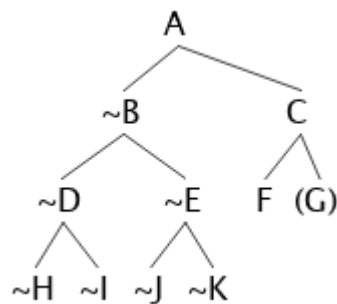
- A otimalidade é garantida, porém, aumenta a complexidade temporal em relação à busca em largura porque tem que explorar os dois caminhos que levam à Bucharest para decidir qual é o mais barato.

### RESPOSTA Q3.

#### BUSCA EM PROFUNDIDADE

**x<Nó>**: nó não incluído na fronteira porque <Nó>.estado já foi explorado  
**-<Nó>**: nó não incluído na fronteira porque <Nó>.estado já está na fronteira e é melhor ou nó substituído na fronteira porque <Nó>.estado gerado é melhor  
**~<nó>**: nó removido da árvore por não ser mais necessário  
**(<Nó>)**: nó que representa a solução

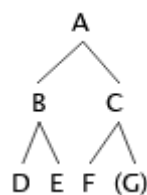
Neste exemplo, não foi considerada a geração de nós já explorados (ex. A como filho de B).



[A[~B[~D[~H][~I]][~E[~J][~K]][C[F][G]]]]

Na busca em profundidade, neste caso particular, foram gerados todos os ramos a partir do nó A até as folhas da subárvore B, porém, no máximo 7 nós foram armazenados na memória: (ABCDEHI) = raiz + b + b + b = 1 + b.m = 1 + 3.2 = 7 (b = fator de ramificação; m=tamanho máximo de caminho no grafo = 3)

#### BUSCA EM LARGURA



[A [B[D][E]][C[F][G]]]]

Neste caso particular, a busca em largura necessitou manter 7 nós em memória. Embora seja o mesmo número que o da busca em profundidade, nota-se que a complexidade cresce de forma distinta aqui: todos os nós que estão na mesma profundidade (d=2) do objetivo devem ser gerados no pior caso (o último nó visitado é o nó objetivo):  $1 + 2^1 + 2^2 = \text{raiz} + \text{nós nível 1} + \text{nós nível 2} = 7$ . Caso o nó objetivo fosse o 'O' então o algoritmo manteria em memória 15 nós enquanto a busca em profundidade manteria os mesmos 7 nós.

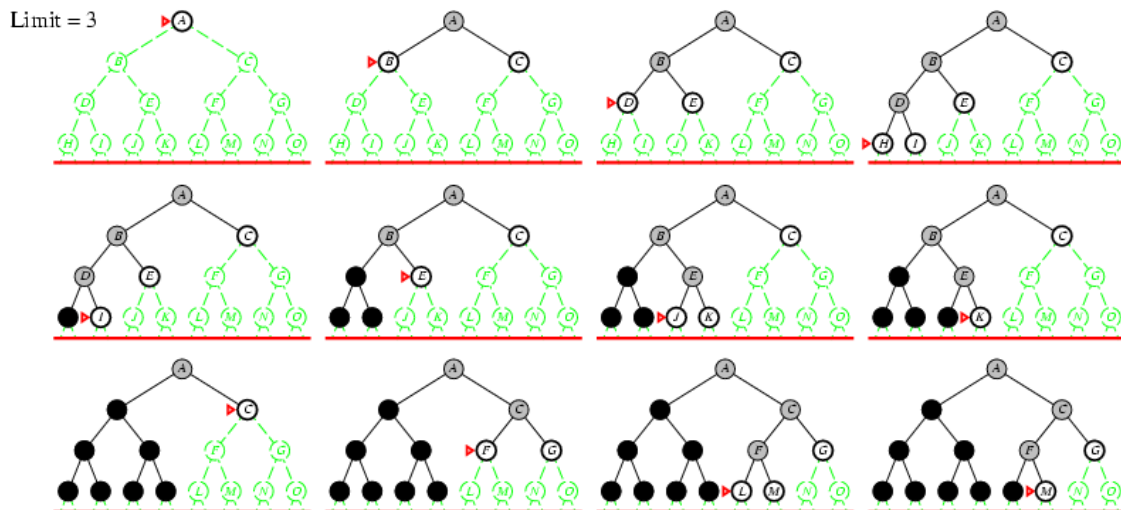
## RESPOSTA Q4

Neste caso, a busca em profundidade tem complexidade temporal exponencial em relação ao caminho máximo do grafo no pior dos casos =  $O(b^m)$  – ex. se o nó objetivo fosse o 'O', todos os nós seriam gerados. A busca em profundidade tem que ir até a profundidade máxima em todos os ramos por isso  $b^m$  enquanto a busca em largura explora somente até a profundidade do nó objetivo mais raso (d).

No caso particular do exercício, que não é o pior caso, foram gerados 11 nós (todos, exceto L, M, N e O).

## RESPOSTA Q5

### COMPLEXIDADE TEMPORAL



Na figura acima, temos a seguinte complexidade temporal – vamos supor que a solução está em d=3 (pior caso nó 'O'):

Level	Nós gerados	Contagem	Polinômios
0	A	1	1
1	ABC	1 + 2	1 + b
2	ABCDEFG	1 + 2 + 4	1 + b + b <sup>2</sup>
3	ABCDEFGHIJKLMNO	1 + 2 + 4 + 8	1 + b + b <sup>2</sup> + b <sup>3</sup>

Se observarmos os polinômios por colunas vemos que:

- o nó raiz de profundidade 0 foi gerado 4 vezes = (d+1)
- os nós de profundidade 1 foram gerados 3 vezes = (d)
- os nós de profundidade 2 foram gerados 2 vezes = (d-1)
- os nós de profundidade 3 foram gerados 1 vez = (d-2)

Observar que para grandes árvores, o fato de repetirmos a geração de nós de profundidade inferior à  $d$  não tem um impacto grande no total de nós gerados. Vamos supor que temos  $b=12$ ,  $d=5$  e ao atingirmos a igualdade  $l=d$  a busca em aprofundamento iterativo terá gerado:

	PROFUNDIDADE ( $d=5$ )					
ITERAÇÃO	0	1	2	3	4	5
0	1					
1	1	12				
2	1	12	144			
3	1	12	144	1.728		
4	1	12	144	1.728	20.736	
5	1	12	144	1.728	20.736	248.832
Nós gerados por prof.	6	60	576	5.184	41.472	248.832
Nós gerados	296.130	aprofundamento iterativo				
Nós gerados	271.453	busca em profundidade				
diferença	24.677					
diferença %	109%					

### COMPLEXIDADE ESPACIAL

É similar à da busca em profundidade, basta manter o ramo em exploração na memória e mais alguns nós que ainda não tiveram todos os filhos explorados. Para um limite  $l$  qualquer, a complexidade é  $O(b \cdot l)$ .