Disciplina: Inteligência Artificial

Professora: Cristiane Neri Nobre

Data de entrega: 20/10

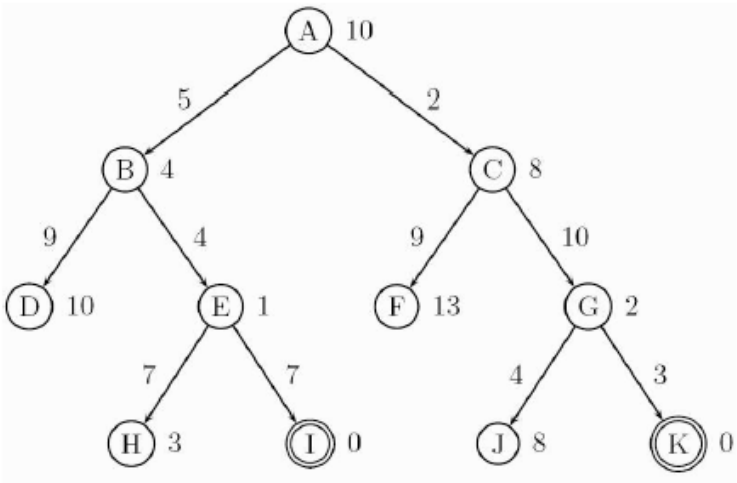
Aluno : Henrique Oliveira da Cunha Franco

Questão 01

Considere o espaço de busca a seguir. Cada nó é rotulado por uma letra. Cada nó objetivo é representado por um círculo duplo. Existe uma heurística estimada para cada dado nó (indicada por um valor ao lado do nó). Arcos representam os operadores e seus custos associados. Para cada um dos algoritmos a seguir, pede-se:

1) Os **nós visitados** na ordem em que eles são examinados, começando pelo nó A
2) Forneça também a **solução obtida** por cada método
3) Pergunta-se: a **heurística** é admissível? Justifique.

No caso de escolhas equivalentes entre diferentes nodos, prefira o nodo mais próximo da raiz, seguido pelo nodo mais à esquerda na árvore. O algoritmo pára a busca quando encontra o I ou o K. Ou seja, não é necessário encontrar os dois objetivos.



1) Algoritmo de Busca em Largura
2) Algoritmo de Busca em Profundidade
3) Custo Uniforme
4) Algoritmo de Busca Gulosa
5) Algoritmo A*

Questão 02

Para o problema do Puzzle de 8, pede-se:

1. A heurística de Manhattan é admissível? Justifique.
2. Proponha uma outra heurística para este problema. Ela é admissível? Justifique.

Lista 4 - Inteligência Artificial

Henrique Oliveira da Cunha Franco

Questão 01

Algoritmo de Busca em Largura (BFS)

1. Nós Visitados:
A BFS explora a árvore nível por nível, começando pelo nó A, visitando seus filhos B e C, em seguida os filhos desses nós, e assim por diante.

- Ordem: $A, B, C, D, E, F, G, H, I, J, K, O$

2. Solução Obtida:
A BFS encontra o objetivo O antes de K.

- Caminho: $A \to C \to G \to K$

3. Heurística:
A BFS não utiliza heurística, sendo uma busca cega. Portanto, a pergunta sobre admissibilidade não se aplica.

Algoritmo de Busca em Profundidade (DFS)

1. Nós Visitados:
A DFS explora o mais profundamente possível antes de retroceder. Começa pelo nó A e segue pela esquerda.

- Ordem: $A, B, D, E, H, I, C, F, G, J, K$

2. Solução Obtida:
A DFS encontra o objetivo K seguindo um caminho mais profundo.

- Caminho: $A \to C \to G \to K$

3. Heurística:
A DFS também não utiliza heurística, então a admissibilidade não se aplica.

Algoritmo de Custo Uniforme

1. Nós Visitados:
A busca de custo uniforme expande os nós com o menor custo acumulado primeiro.

- Ordem: A, C, B, G, E, F, J, K

2. Solução Obtida:
A busca de custo uniforme encontra K antes de O, já que K tem um custo acumulado mais baixo.

- Caminho: $A \to C \to G \to K$

3. Heurística:
Não há heurística envolvida, já que a busca é guiada pelo custo real. Admissibilidade não se aplica.

Algoritmo de Busca Gulosa

1. Nós Visitados:
A busca gulosa escolhe o nó com a menor heurística (valor ao lado do nó).

- Ordem: A, C, G, K

2. Solução Obtida:
A busca gulosa encontra K rapidamente, dado que a heurística do nó K é zero.

- Caminho: $A \to C \to G \to K$

3. Heurística:
A heurística é admissível se nunca superestima o custo real para o objetivo. Neste caso, a heurística parece admissível, pois os valores dos nós são consistentes com os custos reais (nunca superestimam).

Algoritmo A*

1. Nós Visitados:
O A* combina o custo acumulado e a heurística, expandindo os nós com o menor valor total.

- Ordem: A, C, G, K

2. Solução Obtida:
O algoritmo A* também encontra K, pois esse nó tem o menor valor total somando custo e heurística.

- Caminho: $A \to C \to G \to K$

3. Heurística:
No A*, a admissibilidade da heurística é essencial para garantir a otimização. Como a heurística não superestima o custo real, ela é admissível.

Questão 2

1.

A heurística de Manhattan é a soma das distâncias horizontais e verticais de cada peça até sua posição objetivo no tabuleiro. Ou seja, ela calcula o número total de movimentos que cada peça deve fazer para chegar à posição correta, ignorando outros obstáculos.

Admissibilidade: A heurística de Manhattan é admissível, pois nunca superestima o número de movimentos reais necessários para resolver o puzzle. Em outras palavras, a distância de Manhattan fornece um valor que é sempre menor ou igual ao custo real para mover cada peça para sua posição final. Como a admissibilidade implica que a heurística deve ser otimista (subestimar ou ser igual ao custo real), a heurística de Manhattan é admissível.

2.

Heurística Proposta: Uma outra heurística possível é a *quantidade de peças fora de posição*, que conta quantas peças não estão na sua posição correta no tabuleiro.

Admissibilidade: Esta heurística é admissível, pois o número de peças fora de posição nunca superestima o número de movimentos reais necessários para resolver o puzzle. No entanto, ela é uma heurística mais fraca que a heurística de Manhattan, já que, ao contar apenas o número de peças fora de posição, ela pode subestimar em muito o custo real (por exemplo, uma peça pode estar fora de posição, mas pode exigir muitos movimentos para ser colocada corretamente). Portanto, enquanto a heurística é admissível, ela é menos informativa do que a heurística de Manhattan.

Questão 03

Julgue os itens a seguir, relativos a métodos de busca com informação (busca heurística) e sem informação (busca cega), aplicados a problemas em que todas as ações têm o mesmo custo, o grafo de busca tem fator de ramificação finito e as ações não retornam a estados já visitados.

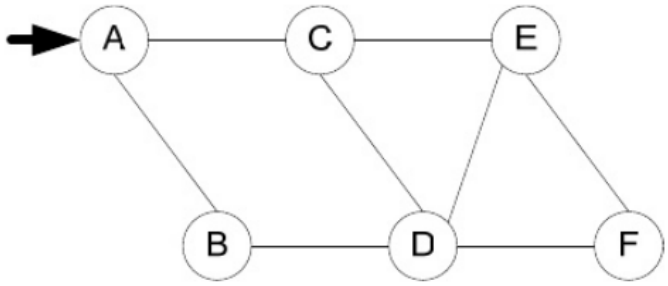
I. A primeira solução encontrada pela estratégia de busca em largura é a solução ótima.
II. A primeira solução encontrada pela estratégia de busca em profundidade é a solução ótima.
III. As estratégias de busca com informação usam funções heurísticas que, quando bem definidas, permitem melhorar a eficiência da busca.
IV. A estratégia de busca gulosa é eficiente porque expande apenas os nós que estão no caminho da solução.

Estão certos apenas os itens

X a) I e II.
b) I e III.
c) I e IV.
d) II e IV.
e) III e IV.

Questão 04

Considere o algoritmo de busca em largura em grafos. Dado o grafo a seguir e o vértice A como ponto de partida, a ordem em que os vértices são descobertos é dada por:



X A) A B C D E F
B) A B D C E F
C) A C D B F E
D) A B C E D F
E) A B D F E C

Questão 05

Analise as seguintes as seguintes afirmativas:

I. A estratégia de busca em largura encontra a solução ótima quando todos os operadores de mudança de estado têm o mesmo custo.

II. A estratégia de busca em profundidade sempre expande um menor número de nós que a estratégia de busca em largura, quando aplicadas ao mesmo problema.

III. A estratégia de busca heurística encontra sempre a solução de menor custo.

IV. A estratégia de busca heurística expande um número de nós em geral menor que o algoritmo de busca em largura, mas não garante encontrar a solução ótima.

V. O algoritmo de busca heurística que utiliza uma função heurística admissível encontra a solução ótima.

A esse respeito, pode-se concluir que

(a) apenas a afirmativa V é correta.

(b) todas as afirmativas são corretas.

(c) todas as afirmativas são falsas.

(d) apenas as afirmativas II e V são corretas.

(X) apenas as afirmativas I, IV e V são corretas.

Questão 06 - POSCOMP 2007

[**TE**] Considerando que $h(n)$ é o custo estimado do nó n até o objetivo, em relação à busca informada, pode-se afirmar que

(a) a busca *gulosa* minimiza $h(n)$.

(b) a busca A^* minimiza $h(n)$.

(X) a busca de custo uniforme minimiza $h(n)$.

(d) a busca *gulosa* minimiza $h(n)$ somente se a heurística for admissível.

(e) a busca A^* minimiza $h(n)$ somente se a heurística for admissível.

Questão 07 - POSCOMP 2005

Considere $h(x)$ como uma função heurística que define a distância de x até a meta; considere ainda $h^r(x)$ como a distância real de x até a meta. $h(x)$ é dita admissível se e somente se:

(a) $\exists n\ h(n) \leq h^r(n)$.

(X) $\forall n\ h(n) \leq h^r(n)$.

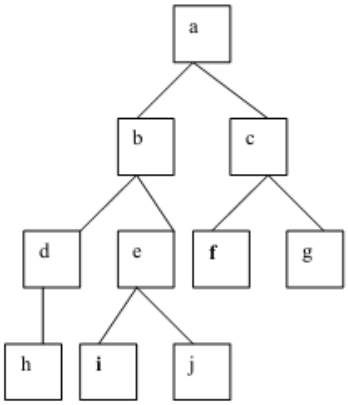
(c) $\forall n\ h(n) > h^r(n)$.

(d) $\exists n\ h(n) > h^r(n)$.

(e) $\exists n\ h(n) < h^r(n)$.

Questão 8

59. Seja a árvore binária abaixo a representação de um espaço de estados para um problema p, em que o estado inicial é a, **e** i **e** f **são estados finais.**



Um algoritmo de busca em largura-primeiro forneceria a seguinte seqüência de estados como primeira alternativa a um caminho-solução para o problema p:

a) a b d h e i

X) a b c d e f

c) a b e i

d) a c f

e) a b d e f

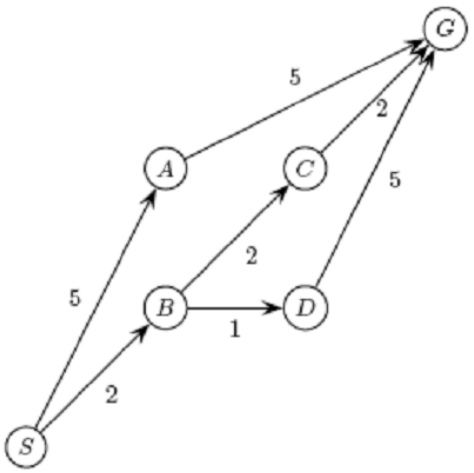
Questão 9

Suponha um algoritmo de busca pelo melhor primeiro (best-first ou busca gulosa) em que a função objetivo é f(n) = (2 – w).g(n) + w.h(n). Que tipo de busca ele realiza quando w = 0? Quando w = 1? E quando w = 2?

Questão 10

Considere o espaço de busca abaixo, onde S é o estado inicial e G é o único estado que satisfaz o teste de objetivo. Os rótulos nas arestas indicam o custo de percorrê-las e a tabela ao lado mostra o valor de

três heurísticas h1, h2 e h3 para cada estado.



Node	h_0	h_1	h_2
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

Questão 9

Suponha um algoritmo de busca pelo melhor primeiro (best-first ou busca gulosa) em que a função objetivo é:

$$f(n) = (2 - w) \cdot g(n) + w \cdot h(n)$$

onde $g(n)$ é o custo acumulado até o nó n e $h(n)$ é a heurística estimada a partir de n. Vamos analisar o comportamento do algoritmo para diferentes valores de w.

Caso em que $w = 0$

Quando $w = 0$, a função objetivo se torna:

$$f(n) = (2 - 0) \cdot g(n) + 0 \cdot h(n) = 2 \cdot g(n)$$

Neste caso, o algoritmo considera apenas o custo acumulado $g(n)$, o que significa que ele realiza uma **busca de custo uniforme**. Essa busca expande os nós com o menor custo acumulado até o momento, ignorando qualquer estimativa heurística.

Caso em que $w = 1$

Quando $w = 1$, a função objetivo se torna:

$$f(n) = (2 - 1) \cdot g(n) + 1 \cdot h(n) = g(n) + h(n)$$

Neste caso, o algoritmo realiza uma busca **A (A-estrela)**. O valor de $f(n)$ é a soma do custo acumulado $g(n)$ e da estimativa heurística $h(n)$. A busca A é ótima e completa se a heurística for admissível, ou seja, se nunca superestimar o custo real até o objetivo.

Caso em que $w = 2$

Quando $w = 2$, a função objetivo se torna:

$$f(n) = (2 - 2) \cdot g(n) + 2 \cdot h(n) = 2 \cdot h(n)$$

Neste caso, o algoritmo realiza uma **busca gulosa**. Ele ignora completamente o custo acumulado $g(n)$ e faz escolhas baseadas apenas na heurística $h(n)$. Esse modelo de busca não garante encontrar o caminho mais curto, já que é possível que a heurística leve o algoritmo a expandir nós que estão mais longe do objetivo em termos de custo real.

1) Em relação à busca A*, pede-se:
    a) Quais são os nós expandidos pela busca A* usando cada uma das heurísticas (h1, h2 e h3)?
    b) Qual é a solução (caminho) encontrado por cada uma delas?
    c) Quais das heurísticas são admissíveis? Justifique sua resposta.

2) Em relação à busca gulosa, pede-se:
    a) Qual são os nós expandidos?
    b) Qual é a solução (caminho) encontrado?

3) Em relação à busca em profundidade, pede-se:
    c) Qual são os nós expandidos?
    d) Qual é a solução (caminho) encontrado?

4) Em relação à busca em largura, pede-se:
    e) Qual são os nós expandidos?
    f) Qual é a solução (caminho) encontrado?

Questão 11

Considere um jogo do tipo 8-puzzle, cujo objetivo é conduzir o tabuleiro esquematizado na figura abaixo para o seguinte estado final.

1	2	3
8		4
7	6	5

Considere, ainda, que, em determinado instante do jogo, se tenha o estado E0 a seguir.

3	4	6
5	8	
2	1	7

Pelas regras desse jogo, sabe-se que os próximos estados possíveis são os estados E1, E2 e E3 mostrados abaixo.

Questão 10

1 - Busca A*

a) Nós expandidos pela busca A* usando cada uma das heurísticas (h1, h2 e h3)

- Usando h_1:
  - Nós expandidos: S, A, C, G
- Usando h_2:
  - Nós expandidos: S, B, D, G
- Usando h_3:
  - Nós expandidos: S, C, G

b) Caminho encontrado por cada uma das heurísticas

- Usando h_1: $S \to A \to C \to G$
- Usando h_2: $S \to B \to D \to G$
- Usando h_3: $S \to C \to G$

c) Heurísticas admissíveis

Uma heurística é admissível se nunca superestima o custo real para atingir o objetivo.

- h_1: Admissível, pois em todos os nós o valor da heurística nunca excede o custo real para alcançar o objetivo.
- h_2: Admissível, pelos mesmos motivos que h_1. Não superestima o custo real em nenhum nó.
- h_3: Não é admissível, pois no nó C, o valor da heurística (5) é maior do que o custo real para alcançar o objetivo (2).

2 - Busca Gulosa

a) Nós expandidos

A busca gulosa usa apenas a função heurística $h(n)$ para escolher o próximo nó a expandir.

- Usando h_1: Nós expandidos: S, A, C, G
- Usando h_2: Nós expandidos: S, B, D, G
- Usando h_3: Nós expandidos: S, C, G

b) Caminho encontrado

- Usando h_1: $S \to A \to C \to G$
- Usando h_2: $S \to B \to D \to G$
- Usando h_3: $S \to C \to G$

3 - Busca em Profundidade

c) Nós expandidos

A busca em profundidade explora o caminho mais profundo antes de retroceder.

- Nós expandidos: S, A, C, G

d) Caminho encontrado

- Caminho: $S \to A \to C \to G$

4 - Busca em Largura

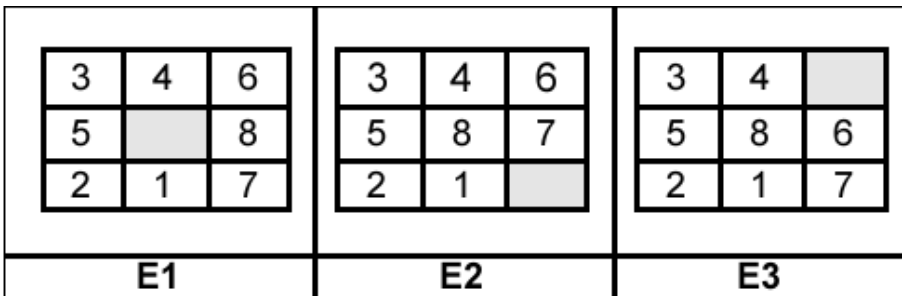
e) Nós expandidos

A busca em largura expande nós nível por nível.

- Nós expandidos: S, A, B, C, D, G

f) Caminho encontrado

- Caminho: $S \to A \to C \to G$



Considere uma função heurística **h** embasada na soma das distâncias das peças em relação ao estado final desejado, em que a distância **d** a que uma peça **p** está da posição final é dada pela soma do número de linhas com o número de colunas que a separam da posição final desejada.

Por exemplo, em E1, **d(1) = 2 + 1 = 3**. A partir dessas informações analise as asserções a seguir.

Utilizando-se um algoritmo de busca gulosa pela melhor escolha que utiliza a função h, o próximo estado no desenvolvimento do jogo a partir do estado E0 tem de ser E3

porque,

dos três estados E1, E2 e E3 possíveis, o est ado com menor soma das distâncias entre a posição atual das peças e a posição final é o estado E3.

Assinale a opção correta a respeito dessas asserções.

a) As duas asserções são proposições verdadeiras, e a segunda é uma justificativa correta da primeira.
b) As duas asserções são proposições verdadeiras, e a segunda não é uma justificativa correta da primeira.
c) A primeira asserção é uma proposição verdadeira, e a segunda é uma proposição falsa.
d) A primeira asserção é uma proposição falsa, e a segunda é uma proposição verdadeira.
e) As duas asserções são proposições falsas.

Questão 12

Considere um espaço de estados onde o estado inicial é o número 1 e a função sucessor para o estado *n* retorna dois estados, com os números *2n* e *2n+1*.

a. Desenhe a porção do espaço de estados correspondente aos estados 1 a 15.

b. Suponha que o estado objetivo seja 11. Liste a ordem em que os nós serão visitados no caso da busca em extensão, da busca em profundidade limitada com limite 3 e da busca por aprofundamento iterativo.

Questão 13

Investigue vantagens e desvantagens do algoritmo A*.

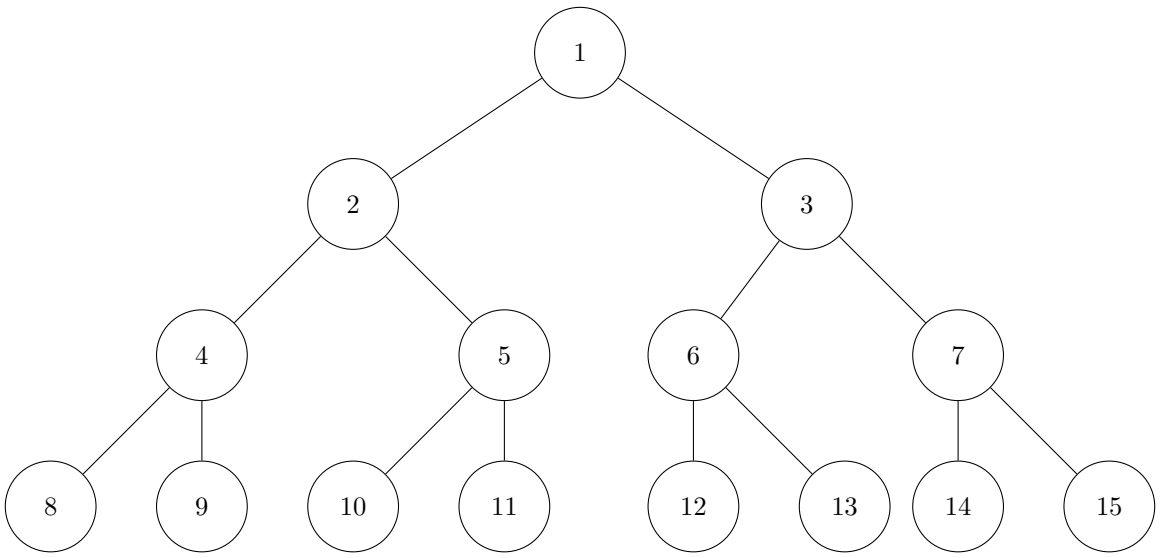
Questão 14

Investigue outros algoritmos que são melhoria do algoritmo A*

Questão 12

Parte (a) - Espaço de Estados

O gráfico abaixo representa os estados de 1 a 15, gerados pela função sucessora:



Parte (b) - Ordens de Visitação

Busca em Largura (Extensão)

A busca em largura visita os nós na seguinte ordem:

$$1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11$$

Busca em Profundidade Limitada (Limite = 3)

Com o limite de profundidade igual a 3, a ordem de visitação será:

$$1, 2, 4, 8, 9, 5, 10, 11$$

Busca por Aprofundamento Iterativo

Na busca por aprofundamento iterativo, os nós são visitados em camadas de profundidade crescente:

- Profundidade 0: 1
- Profundidade 1: $1, 2, 3$
- Profundidade 2: $1, 2, 3, 4, 5, 6, 7$
- Profundidade 3: $1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11$

Assim, a ordem final de visitação é:

$$1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11$$

Questão 13

Vantagens

- **Completo e Ótimo:** Encontra a solução ótima se a heurística for admissível.
- **Flexível:** Pode ser ajustado com diferentes heurísticas para diversos problemas.

Desvantagens

- **Uso elevado de memória:** Armazena todos os nós gerados, o que limita seu uso em grandes problemas.
- **Dependência da heurística:** Se mal escolhida, pode se tornar ineficiente.

Questão 14

IDA*

Reduz o uso de memória ao custo de reexplorar nós, usando profundidade iterativa.

MA* e RBFS

Controlam a memória limitando a expansão de nós, mas podem aumentar o tempo de execução devido à reexpansão.

SMA*

Gerencia melhor o uso de memória e ajusta-se aos recursos disponíveis, mas também pode reexplorar nós.

Theta*

Gera trajetórias mais curtas e suaves, especialmente útil em robótica e jogos, mas pode ser mais custoso de implementar.

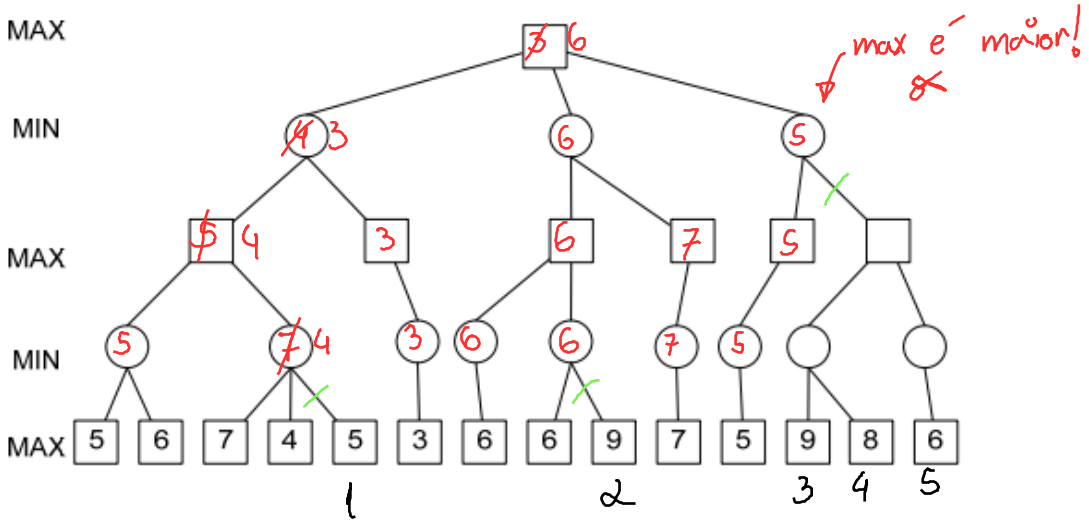
De modo geral, esses algoritmos são variações que tentam equilibrar as limitações de memória e tempo, tornando o A* mais aplicável a problemas práticos em grandes espaços de busca.

Questão 15

Considere a seguinte situação: Dados 5 palitos, cada jogador pode retirar 1, 2 ou 3 por turno. Perde o jogador que retira o último palito. Utilize a busca MINIMAX para verificar se MAX pode ganhar o jogo.

Questão 16

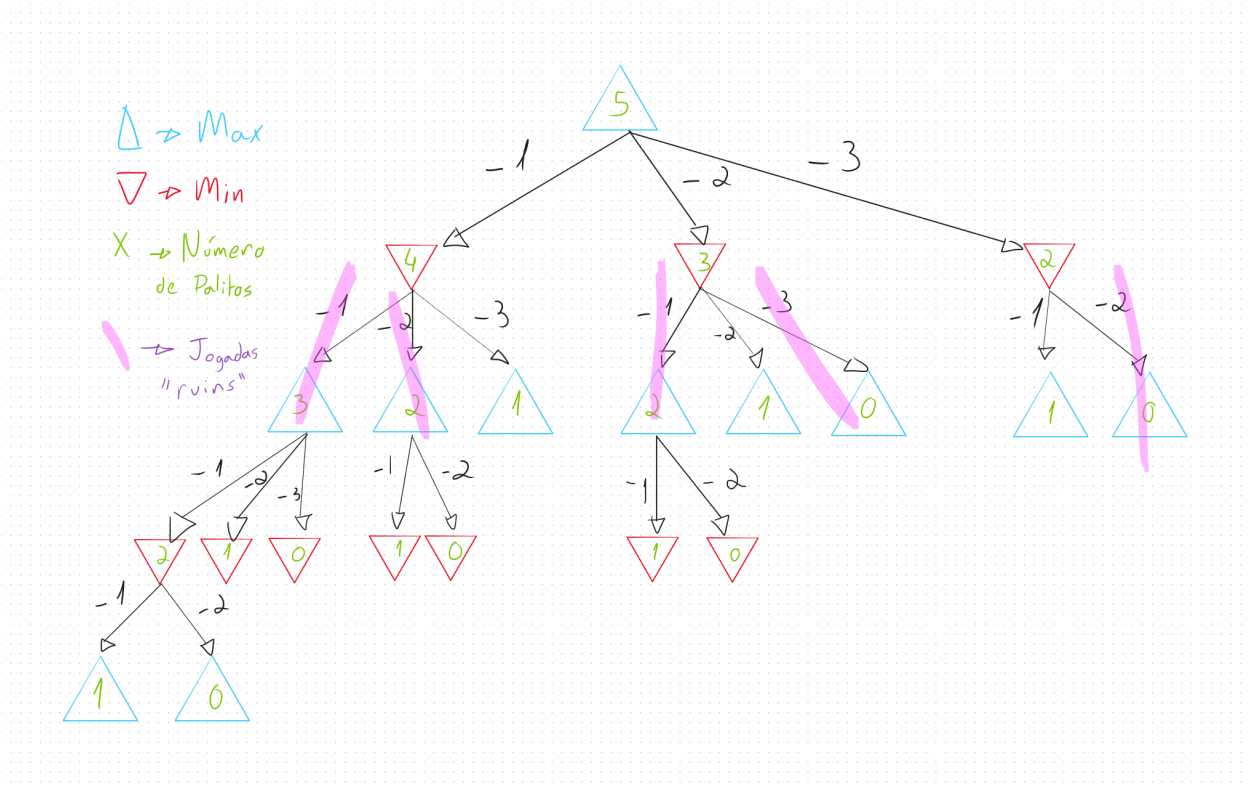
Considere a árvore minimax abaixo, representando um jogo onde queremos maximizar o valor da função de avaliação estática:



Assinale a alternativa que apresenta a quantidade de folhas que não deverão ser visitados em uma busca da melhor jogada se a estratégia de **poda alfa-beta** for utilizada.

a) 5
b) 8
c) 9
d) 10
e) 11

Questão 15



Ao fazer um esquema-árvore das jogadas possíveis nesse dado jogo, Max não pode ganhar esse jogo, pois independente da quantidade da palitos que ele retirar, (supondo que o outro jogador fará as melhoras jogadas possíveis), Mini ainda poderá retirar 4 palitos da quantidade total restante (seja começando por 1, então tirando 3, começando por 2, depois tirando 2, e começando por 3, depois tirando 1), obrigando Max a retirar o último palito do jogo, fazendo-o perdê-lo.