

HEURÍSTICAS (H)

A HR (Heurística Resultante) começa com valor diferente de 0, para evitar multiplicações e divisões nulas, além da divisão de denominador nulo.

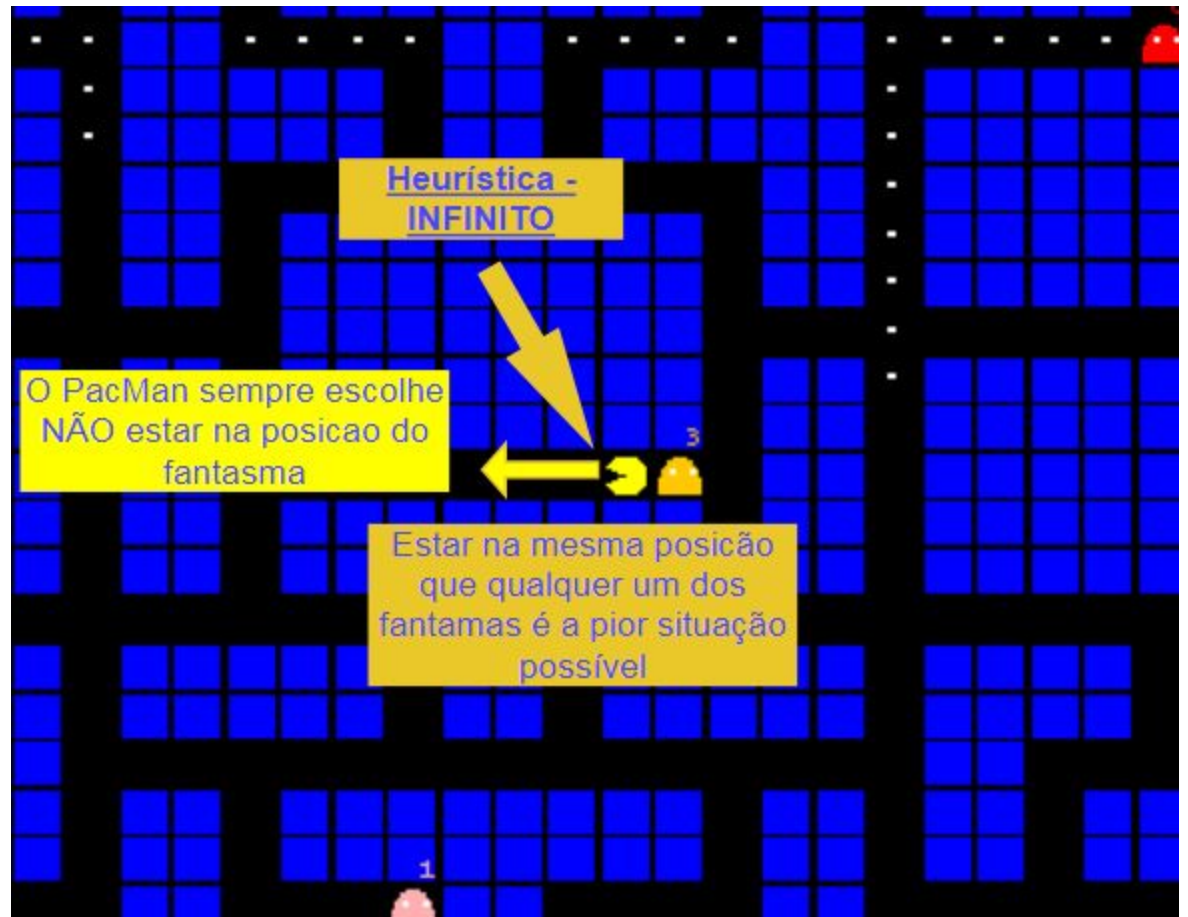
As heurísticas relativas à distância são testadas com a dist. Euclidiana (e) e de Manhattan(m)

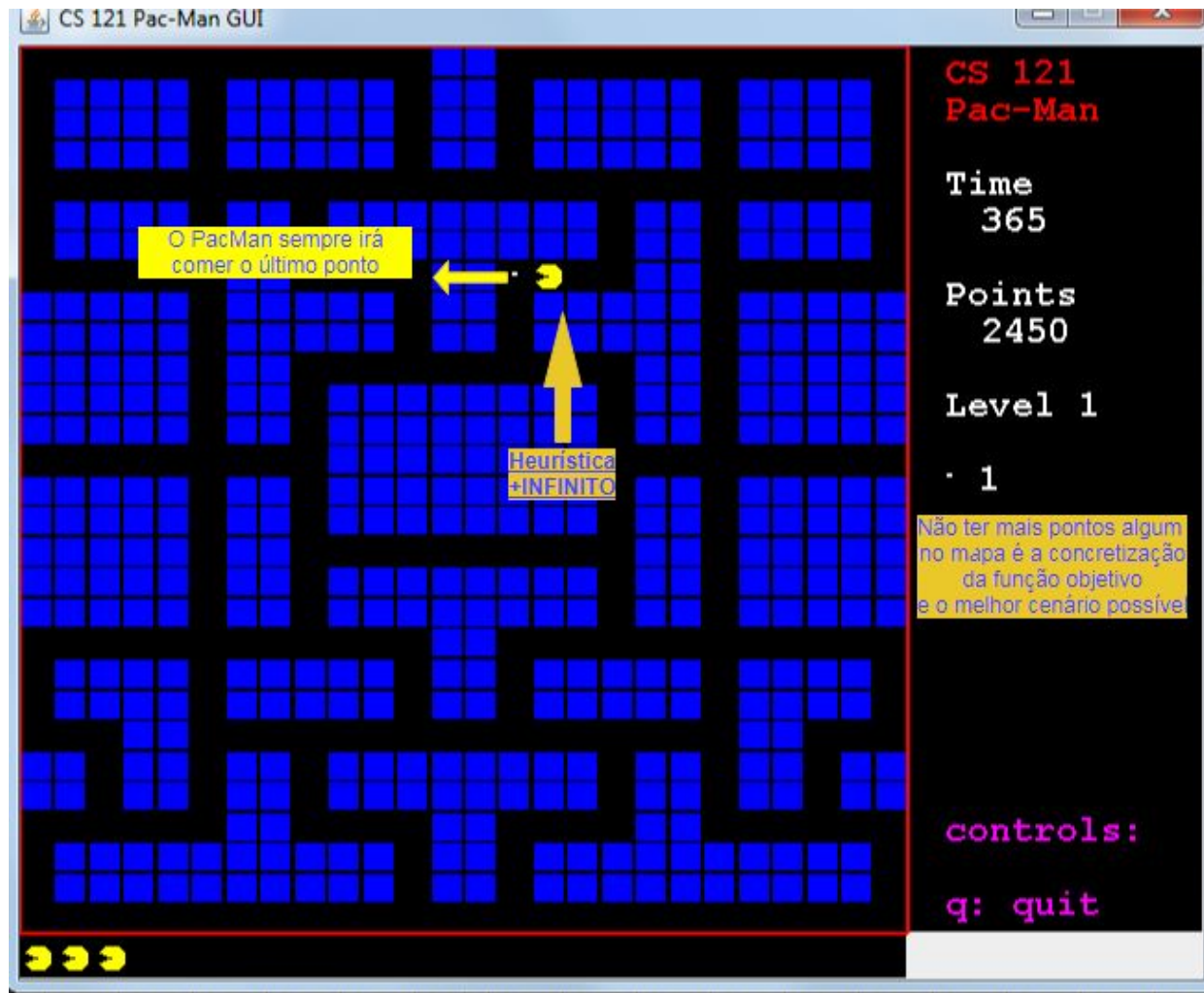
0: FUNÇÃO OBJETIVO

Se resultado = Vitória, HR é somado de +Infinito;

Se resultado = Derrota, HR é somado de - Infinito

Preferencialmente, a I.A. segue o caminho favorecido pela primeira condição

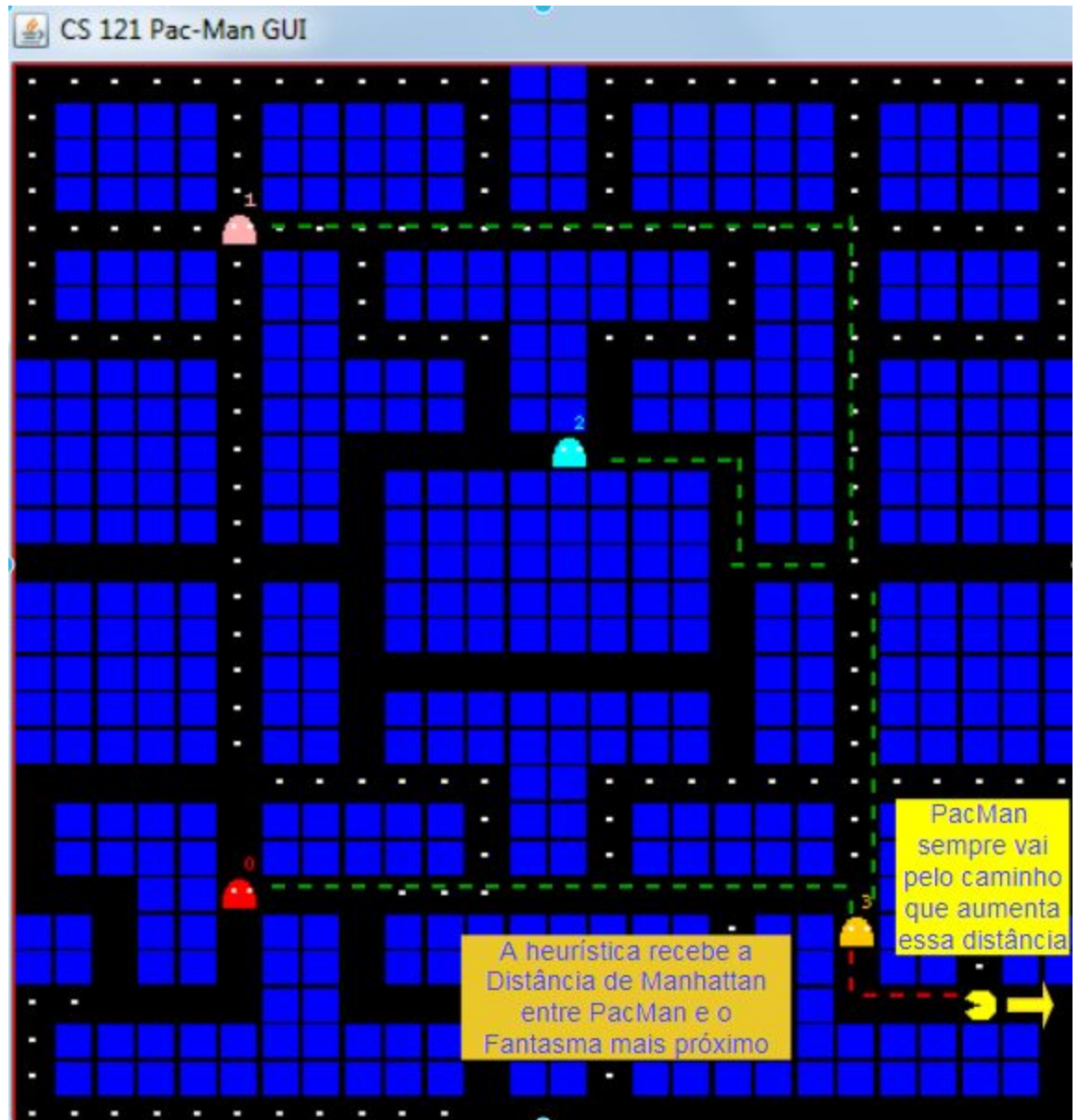




1: DISTÂNCIA DE MANHATTAN ENTRE PACMAN E O FANTASMA MAIS PRÓXIMO

A HR é somada desta heurística sem qualquer outro peso.

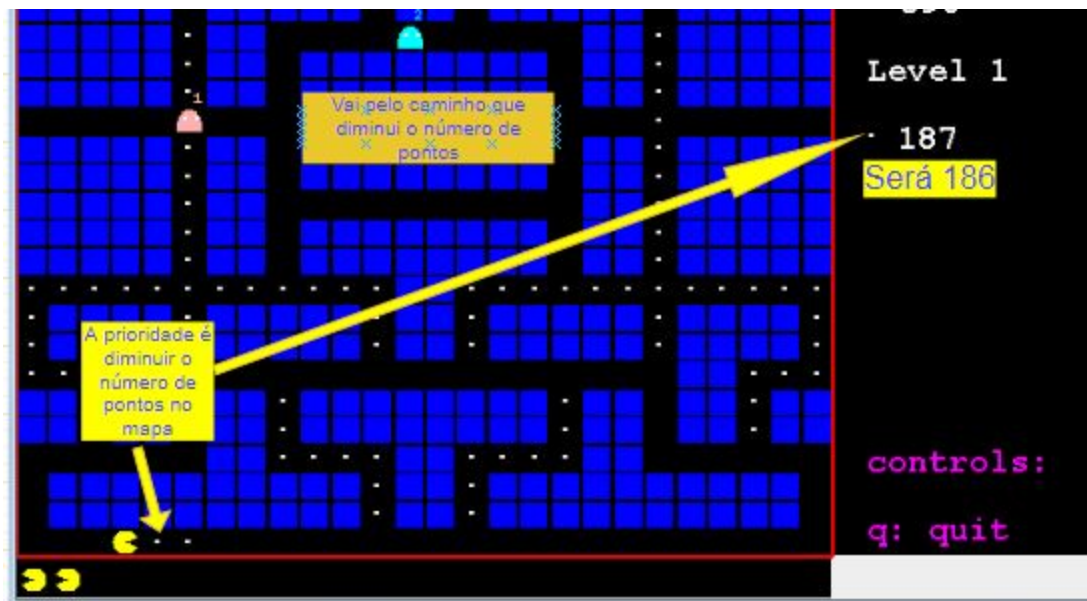
Preferencialmente a .I.A. opta pelo caminho mais distante do fantasma mais próximo



2: QUANTIDADE DE PONTOS EXISTENTES NO MAPA

A HR é subtraída dessa heurística, portanto quanto maior ela for, menor a HR será.

A I.A. opta pelo caminho que resultará na diminuição do total de pontos presentes no mapa, assim favorecendo sua vitória.



3: DISTÂNCIA DE MANHATTAN ENTRE PACMAN E O PONTO MAIS PRÓXIMO

A HR é subtraída dessa H

A I.A. optará por ir pelo caminho que a deixe mais perto do ponto mais próximo



ALGORITMOS DE BUSCA (A)

BUSCA GULOSA (Greedy)

Abre sempre o nó seguinte, escolhendo aquele com a maior HR.

BUSCA EM LARGURA LIMITADA (BFS)

Expande a árvore de possibilidades até um nível de profundidade manualmente estabelecido e usa o último nível desta árvore para decidir o próximo movimento a ser executado

CONTRA-MÉTODO PARA LOOPING (CML)

A: MOVIMENTO NULO (Usado na Busca Gulosa)

O estado atual também é avaliado, e caso nenhum movimento resulte em uma melhor HR, a I.A. simplesmente permanece imóvel

B: MOVIMENTO DE RETORNO = MOVIMENTO ILEGAL (Usado no BFS)

O PacMan é impedido de abrir o nó que o levaria há uma posição igual à anterior.

Um movimento reverso é sempre ilegal, mesmo que a outra opção seja a derrota.

```
if (lastMove != Move.NONE && i == 0) {  
    int backwardNode_index = 0;  
    for (Node node1 : allNodes.get(1)) {  
        if (node1.getMove().equals(lastMove.getOpposite()))  
            backwardNode_index = allNodes.get(1).indexOf(node1);  
    }  
    allNodes.get(1).remove(backwardNode_index);  
}
```



CONCLUSÕES

O melhor em absoluto foi: $H0+H2$ em $BFS<Depth\ 9>$, que conseguiu chegar no 3o nível

- Greedy
 - Melhor combinação sem pesos: $H0+H1m+H2$
 - Pior combinação sem pesos: $H1e+H3e$

Das heurísticas implementadas:

- A $H0$ se provou redundante se ativada junto com $H1$, mas as duas separadas obtiveram resultados diferentes, sendo que $H0$ se provou mais eficaz na coleta de pontos considerando que a ativação da $H1$ faz com que o PacMan obedeça um padrão de ficar em um determinado canto do mapa, acuado pelos fantasmas.
- Das heurísticas, a que mais pareceu dispensável foi a $H3$ (em qualquer uma das distâncias), já que a $H1$ e $H0$ eventualmente levam o PacMan de volta à trilha de pontos, fazendo com que $H2$ atinja seu potencial, enquanto a $H3$ faz isso sacrificando um pouco da cautela da I.A.

Dos diferentes valores de profundidade:

- Conclusões inferidas do BFS
- A progressão dos valores de profundidade não parece ter uma relação linear com as heurísticas, já que diferentes combinações podem ter diferentes efeitos em diferentes profundidades, as vezes até piorando a eficácia da I.A. conforme o aumento da profundidade - Isso acontece especialmente com a ativação da $H1$, já que na atual construção do algoritmo, o PacMan é incapaz de projetar o movimento dos fantasmas, assim projetando seus movimentos levando em consideração à imobilidade dos fantasmas - Nesse caso desempenho piora conforme o aumento de profundidade.
- Quando as outras heurísticas estão ativadas, somente a $H0$ aparenta uma diferença significativa, fornecendo um comportamento parecido com o da $H1$, num esquema de cautela e salvaguarda, mas em um comportamento parecido com a de uma função parabólica, o potencial dessa heurística é atingido no nível 9 de profundidade, e passa a decair após isso. Isso provavelmente acontece pelas dimensões do tabuleiro favorecerem curvas em distâncias de 9 unidades e tornar a previsão cada vez mais ineficaz ao passar disso.
- Quanto maior o nível de profundidade com as $H0$ e $H1$ ativadas, maior a “paranoia” do PacMan, levando-o até mesmo a seguir para a própria derrota.

Dos diferentes pesos:

- Conforme um valor é dividido ou subtraído do HR, maior a tendência dele ser reduzido à inatividade, levando a um comportamento da combinação somente das outras heurísticas.
- Dar maior prioridade à $H1$ costuma levar o PacMan a ficar muito cauteloso, ao ponto de se levar a derrota.

EXPERIMENTOS

H0+H1m+H2+H3m

Greedy

PONTOS: 2020 | Tempo: 551 | Level: 1 | Eficácia: 2571 (PONTOS-Tempo)

A I.A. para quando está a frente de um ponto singular.

BFS <Depth 1>

PONTOS: 4120 | Tempo: 1290 | Level: 2

BFS <Depth 2>

PONTOS: 2440 | Tempo: 975 | Level: 1

BFS <Depth 3>

PONTOS: 1610 | Tempo: 345 | Level: 1

BFS <Depth 4>

PONTOS: 1010 | Tempo: 186

BFS <Depth 8>

PONTOS: 2390 | Tempo: 923 | Level: 1

BFS <Depth 9>

PONTOS: 2380 | Tempo: 1143 | Level: 1

BFS <Depth 10>

PONTOS: 2440 | Tempo: 628 | Level: 1

BFS <Depth 20>

PONTOS: 2380 | Tempo: 1372 | Level: 1

H0+H1m+H1e+H2+H3m+H3e

PONTOS: 890 | Tempo: 809 | Level: 1

H0

Greedy

PONTOS: 1400 | Tempo: 2627 | Level: 1 | Eficácia: -1227

A I.A. Fica indo de um lado ao outro em torno da mesma posição até que um fantasma se aproxime e à faça sair do lugar

BFS <Depth 9>

PONTOS: 930 | Tempo: 283 | Level: 1

H1m

PONTOS: 990 | Tempo: 840 | Level: 1 | Eficácia: 150

A I.A. foge dos fantasmas, somente. Sem procurar os pontos, os coletando por mero acaso. Mesmo essa “cautela” nao impede que o PacMan morra facilmente dado que pode haver um conflito entre as diferentes distâncias

H1e

PONTOS: 2000 | Tempo: 2415 | Level: 1

H1m+H1e

PONTOS: 1640 | Tempo: 2784 | Level: 1

H2

PONTOS: 1240 | Tempo: 130 | Level: 1 | Eficácia: 1110

Por percorrer somente a trilha dos pontos, o PacMan acaba perdendo totalmente sua cautela e fica sem rumo ao não ter um ponto num espaço próximo.

O mais eficaz quando sozinho

H3m

PONTOS: 880 | Tempo: 203 | Level: 1 | Eficácia: 667

A I.A. perde a cautela e “trava” ao encontrar um ponto separado dos outros

H3e

PONTOS: 880 | Tempo: 203 | Level: 1

H3m+H3e

PONTOS: 880 | Tempo: 203 | Level: 1

A classificação entre as Heurísticas unitárias fica:

H1e > H0 > H2 > H1m > H3m

H0 + H1m

Greedy

PONTOS: 990 | Tempo: 840 | Level: 1 | Eficácia: 150

O resultado final é semelhante ao teste de H1.

A H1 parece tornar o H0 redundante e vice-versa

BFS <Depth 9>

PONTOS: 1640 | Tempo: 1745 | Level: 1

H0+H2

Greedy

PONTO: 1950 | Tempo: 2370 | Level: 1

Graças à H0, ele coleta mais pontos antes de ficar sem rumo.

BFS <Depth 1>

PONTOS: 2030 | Tempo: 2030 | Level: 1

BFS <Depth 9>

PONTOS: 6150 | Tempo: 1197 | Level 3

BFS <Depth 20>

PONTOS: 2430 | Tempo: 621 | Level 1

H0+H3m

Greedy

PONTOS: 930 | Tempo: 205 | Level: 1

BFS <Depth 9>

PONTOS: 2320 | Tempo: 785 | Level: 1

H1m+H2

PONTOS: 2320 | Tempo: 1772 | Level: 1

Se saiu melhor que a união total. **Considerando que H1 é redundante para H0, é possível que H3 seja contraefetivo.**

H1m+H3m

PONTOS: 890 | Tempo: 267 | Level: 1

H1m+H3e

PONTOS: 950 | Tempo: 443 | Level: 1

A Distância Euclidiana em H3 parece ser MAIS eficaz

H1e+H3e

PONTOS: 740 | Tempo: 416 | Level: 1

A Distância Euclidiana em H1 parece ser MENOS eficaz

H1e+H3m

PONTOS: 900 | Tempo: 283 | Level: 1

Das quatro cobinacoes acima, a melhor é H1m+H3e, que será usada daqui em diante

H0+H1e+H2+H3e

PONTOS: 1100 | Tempo: 329 | Level: 1

Consideravelmente menos eficaz que H1m e H3m

H0+H1m+H2+H3e

PONTOS: 880 | Tempo: 225 | Level: 1

Ainda menos eficaz

H2+H3e

PONTOS: 1240 | Tempo: 134 | Level:

H2+H3m

PONTOS: 1240 | Tempo: 134 | Level: 1

H1m+H2+H3m

PONTOS: 880 | Tempo: 231 | Level: 1

H0+H2+H3m

Greedy

PONTOS: 1780 | Tempo: 379 | Level: 1

BFS <Depth 9>

PONTOS: 2310 | Tempo: 347 | Level: 1

H0+H1m+H3m

Greedy

PONTOS: 890 | Tempo: 267 | Level: 1

BFS <Depth 9>

PONTOS: 2410 | Tempo: 1358 | Level: 1

H0+H1m+H2

Greedy

PONTOS: 2320 | Tempo: 1772 | Level: 1

igual à H1m+H2

BFS <Depth 9>

PONTOS: 2440 | Tempo: 1833 | Level: 1

Continua igual à H1m+H2

H0+H1m+H2/12+H3m

PONTOS: 1350 | Tempo: 379 | Level: 1

H2 foi modificado por ter maior índice numérico entre todos.

Perdeu sua eficácia em relação ao seu equivalente sem peso

$$H_0 + H_1 m + 2 * H_2 + H_3 m$$

PONTOS: 1930 | Tempo: 420 | Level: 1

Um pouco menor que sua contraparte sem peso e maior contraparte dividida

$$H_0 + H_1 m + H_2 / 2 + H_3 m$$

PONTOS: 970 | Tempo: 215 | Level: 1

Mais ineficaz que sua contraparte dividida por 2

$$H_0 + H_1 m + H_2 / 3 + H_3 m$$

PONTOS: 1230 | Tempo: 394 | Level: 1

$$H_0 + H_1 m + H_2 / 1200 + H_3 m$$

PONTOS: 890 | Tempo: 267 | Level: 1