

Nome: Heitor Saulo Dantas Santos

- Descrição do Problema:

Um grupo de amigos está jogando um jogo chamado stardew valley, que a vida em uma fazenda. Um dos amigos, estudante de ciência da computação, quer construir um algoritmo que ache a forma mais lucrativa de efetuar as plantações, ele tem algumas informações para tentar fazer isso, sendo elas: a quantidade de sementes S disponíveis para plantar, o tempo disponível para a plantação P , e uma quantidade de dinheiro D para comprar as sementes, além é claro, das informações de cada semente, como custo, valor de revenda e tempo para o fruto nascer.

- Entrada:

Na primeira linha: número de casos de teste:

C

Na linha seguinte:

S é o número de tipos de sementes.

P é o período total.

D é o dinheiro disponível.

Nas próximas 3 linhas em sequencia

c é uma lista de tamanho S contendo os custos de plantio para cada tipo de semente em cada local.

v é uma lista de tamanho S contendo os valores de venda esperados para cada tipo de semente em cada local.

t é uma lista de tamanho S contendo os tempos necessários para que cada tipo de semente em cada local esteja pronto para a colheita.

- $0 \leq S \leq 60; 0 \leq P \leq 60; 0 \leq D \leq 1000; 1 \leq c \leq 70; 2 \leq v \leq 140; 1 \leq t \leq 15$

- Saída:

A lista de ações passadas ordenadas da forma específica.

- Tempo Máximo para Cada Caso de Teste: 0,0005 segundo

- Entrada Exemplo 1:

```
1
6 27 480
36 52 51 66 56 59
116 61 51 126 78 85
12 13 13 2 11 3
```

- Saída Exemplo 1:

Lucro máximo possível: 126

- Entrada Exemplo 2:

```
3
42 35 802
39 6 7 32 53 36 57 52 38 62 6 47 38 18 16 29 55 35 42 52 57 2 5 26 31 49 63 5 58 65 61 67 41 39 69 24 5
27 46 13 20 51
39 70 99 111 57 36 121 138 43 114 28 126 78 73 16 29 55 64 42 52 57 28 48 129 60 49 79 117 58 65 61
67 41 132 69 87 69 39 86 111 62 51
4 8 7 12 7 10 1 10 6 3 3 3 8 9 12 11 11 12 2 7 13 14 4 5 11 13 10 1 12 5 14 1 12 7 11 3 2 11 12 5 11 12
48 40 311
49 6 3 60 59 49 17 58 20 20 55 66 64 24 55 36 63 17 33 69 24 2 38 38 24 11 69 53 54 56 27 4 32 1 63 22
49 50 50 39 69 35 6 35 29 61 41 63
```

49 130 43 111 127 67 37 110 116 132 55 66 99 102 55 126 76 94 40 69 41 99 78 78 24 41 91 53 54 79 27
 89 70 118 109 106 49 50 125 71 138 74 9 35 36 71 132 63
 4 8 9 5 13 4 5 14 11 7 11 11 6 13 11 1 14 3 6 8 11 14 12 8 1 3 13 9 11 10 9 14 3 2 5 13 3 7 12 13 14 7 8 3 3
 4 3 2
 15 37 795
 41 30 54 51 45 25 59 6 36 31 7 67 58 52 50
 41 30 60 51 117 118 59 58 127 47 44 67 130 133 50
 9 6 14 3 4 14 3 12 4 12 1 12 11 4 5

- Saída Exemplo 2:

Lucro máximo possível: 138

Lucro máximo possível: 138

Lucro máximo possível: 133

- Objetivos:

Avaliar a capacidade de desenvolver algoritmos para problemas relacionados a problemas de maximização, utilizando técnicas como programação dinâmica.

- Técnicas/Algoritmos Válidos:

Deve ser possível resolver o problema através de algoritmos construídos com técnicas como programação dinâmica que possuem complexidade de ordem $O(n*m)$.

- Técnicas/Algoritmos Inválidos:

Algoritmos que tenham complexidade $O((n*m)!)$ (n representando o tamanho do primeiro parâmetro e m o segundo) não devem ser aceitos devido à forma como a lista foi gerada, sendo assim não serão aceitos algoritmos por força bruta que geram todas as possibilidades por busca exaustiva.

