

Sumário

1	Será que Ele Avança?	2
2	<i>Crossword Puzzle</i>	3
3	Jogos Olímpicos à Vista	5
3.1	Tarefa	5
3.2	Dicas de Implementação	6
3.3	Quanto ao <i>search</i>	6

1 Será que Ele Avança?

Ranoberto e Ranubia são duas alegres rãs adolescentes que moram às margens do rio Cubatão, na Serra Dona Francisca. Ranoberto observou as longas pernas saltadoras de Ranubia e decidiu que quer conhecê-la melhor. Como ele é um pouco tímido e não sabe bem como iniciar a conversa, pensou em convidá-la para um jogo divertido, o que pode facilitar esta paquera. Ranubia gostou do estilo dele e aceitou o convite.

O jogo chama-se “Rã Saltadora”: a partir de posições iniciais eles vão saltar um sobre o outro alternadamente. Ambos, Ranoberto e Ranubia, são capazes de saltar há uma distância horizontal máxima de até **10** unidades, em cada salto simples.

Você recebeu uma lista de posições válidas onde Ranoberto e Ranubia podem se posicionar: $x_1 x_2 \dots x_n$. Como Ranoberto é um cavalheiro deixará para Ranubia o primeiro salto. Ranubia começa inicialmente na posição x_1 e Ranoberto começa inicialmente na posição x_2 ; o objetivo deles é alcançar a posição x_n . Determine o número mínimo de saltos necessários para que cada um, Ranoberto ou Ranubia, alcance o objetivo. Aos dois jogadores não é permitido permanecer na mesma posição ao mesmo tempo (afinal, eles ainda estão se conhecendo...), e em cada salto, o jogador que estiver atrás deve pular por cima do jogador à frente.

Entrada

O arquivo de entrada contém múltiplos casos de teste. Cada linha é um caso de teste e contém uma lista de inteiros $x_1 x_2 \dots x_n$, onde $0 \leq x_1 < x_2 \dots < x_n \leq 1000000$ (aqui era o problema original).

Saída

Para cada caso de teste de entrada, imprima o número total de saltos mínimo necessários para que um dos jogadores, ou Ranoberto ou Ranubia, chegue ao destino. Caso nenhum possa alcançar o destino, imprima -1.

Exemplo de Entrada

```
3 5 9 12 15 17
3 5 9 12 30 40
3 5
0 1 7 8 11 15 19
0 1 10 11 19 20
0 1 10 12 19 20
```

Exemplo de Saída

```
3
-1
0
4
3
-1
```

2 Crossword Puzzle

Consider the crossword puzzle shown in 1

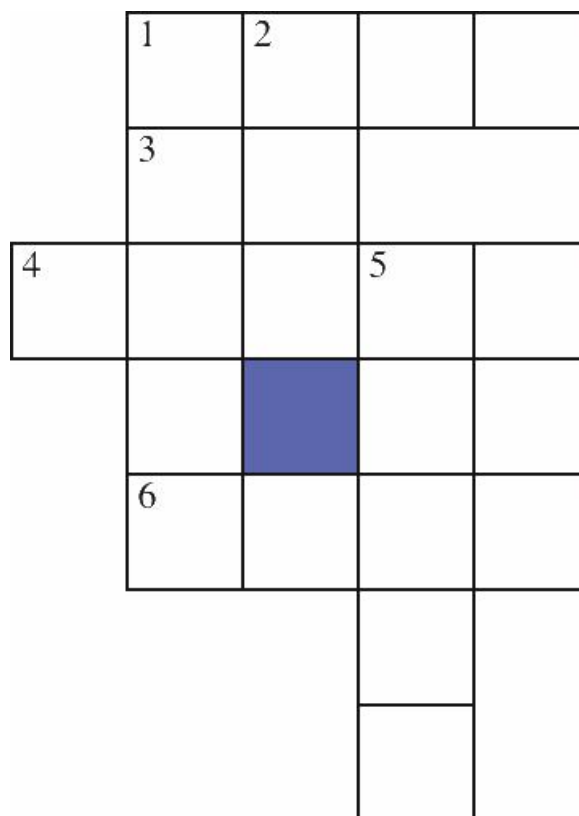


Figura 1: A crossword puzzle to be solved with seven words

The available words that can be used are:

at, eta, be, hat, he, her, it, him, on, one, desk, dance, usage, easy, dove, first, else, loses, fuels, help, haste, given, kind, sense, soon, sound, this, think.

1. Inicialmente dê uma solução para o problema acima, implemente e tenha as respostas.
2. Ok, feita a sua solução que atenda este problema, talvez seja ineficiente para instâncias maiores. Veja por exemplo, algum problema do sítio: <http://rachacuca.com.br/palavras/palavras-cruzadas/>, há muitos similares a esta referência. Logo, pense utilizar alguma restrição global (tipo `table`, `element`, etc), ou criar uma nova restrição para estes problemas, afim de resolver de modo mais eficiente este problema.
3. Em seguida procure responder as questões que se seguem, dada as duas alternativas de implementações que fizeste:
 - (a) Given the representation with nodes for the positions (1-across, 2-down, etc.) and words for the domains, specify the network after domain consistency and arc consistency have halted.
 - (b) Consider the dual representation, in which the squares on the intersection of words are the variables and their domains are the letters that could go in these positions. Give the domains after this network has been made arc consistent. Does the result after arc consistency in this representation correspond to the result in part (a)?

- (c) Show how variable elimination can be used to solve the crossword problem. Start from the arc-consistent network from part (a).
- (d) Does a different elimination ordering affect the efficiency? Explain.

3 Jogos Olímpicos à Vista

Com as proximidades dos Jogos Olímpicos temos a dificuldade dos capitães escolherem os atletas de uma dada modalidade. Assim cada capitão (m) terá que escolher um valor k de atletas de seu plantel, que contém n atletas a serem escolhidos.

Assim, cada capitão deve ter o seu próprio time, tendo dado uma nota sobre o rendimento de cada atleta. Logo, cada técnico deve escolher k atletas, onde no máximo podem diferir por um atleta.

Por exemplo, se tivermos 10 atletas e 3 técnicos, um time pode conter 4 atletas e dois outros com 3 atletas. Não é possível um time com 5, um com 4 e outro com 3.

Assim, admita a seguinte estrutura de dados:

```
n = <number of athletes>;
m = <number of captains>;
array[1..m,1..n] of int: nota;
%nota[c,p] eh o valor/nota dada pelo capitao c para o atleta p
```

Voce deve encontrar um valor máximo como objetivo para uma dada distribuição entre entre os n atletas para os m capitães. Observe que o plantel é único. Veja o exemplo abaixo:

```
n = 10;
m = 3;
nota = [| 4, 5, 6, 7, 2, 1, 0, 4, 5, 6
        | 0, 5, 8, 7, 3, 2, 1, 5, 4, 7
        | 5, 4, 5, 6, 3, 1, 0, 4, 6, 4 |];
```

Veja, o coitado do atleta 7 está mal cotado pelos técnicos 1 e 3, tomou um zero por ambos os técnicos! Provavelmente, este coitado não deve ser selecionado pelo por estes dois técnicos. Já os atletas 3 e 4 estão bem cotados por todos os técnicos, com boas notas.

Assim, a saída para este caso é dada por:

```
obj = 48;
times = [3, 1, 2, 1, 3, 2, 2, 2, 3, 1];
```

Explicando: onde o time 1 (técnico 1) ficou com os seguintes atletas: {2,4,10}. Quanto ao time 2 (do técnico 2) ficou com os seguintes atletas: {3,6,7,8}. O infeliz do atleta 7 ficou mesmo no time 2. Finalmente, o o time 3 (técnico 3) ficou com os seguintes atletas: {1,5,9}. Ou seja, a saída acima é reflexiva aos n atletas onde cada posição indica o atleta em questão.

Claro, que nenhum atleta pode ser selecionado por dois técnicos simultaneamente.

3.1 Tarefa

Implemente este problema, execute para as instâncias dada no diretório de **data**, com arquivos: **teamagain1.dzn**, **teamagain2.dzn**, **teamagain3.dzn**, **teamagain4.dzn**.

A segunda parte é preencher a tabela (vou acertar os nomes em breve – veja abaixo) com os diversos tipos de escolha de variáveis e de domínio. Siga como exemplo a tabela abaixo (referência: tabelas 1 e 2) como guia, e escolha alguns testes do quais voce entende que para este problema vai alcançar um bons e péssimos resultados.

Preencha as tabelas 1 e 2 com cada um dos arquivos testes.

Conclua o experimento, respondendo:

Tabela 1: Tempos de execução com manipulação dos parâmetros do *search*. $N = \dots$

Sel. Variável Atr. Dominio	first_fail	anti_first_fail	input_order
indomain_min
indomain_max
indomain_median
indomain_split
indomain_random
indomain_interval

Tabela 2: Continuação da tabela anterior. $N = \dots$

Sel. Variável Atr. Dominio	largest	smallest	max_regret
indomain_min
indomain_max
indomain_median
indomain_split
indomain_random
indomain_interval

1. Quais as estratégias que ocorreram os piores e melhores resultados para este problema em específico?
2. O que poderia ser feito para ser melhorado?
3. Faça uma modificação para o problema acima, e apresente a saída no seguinte formato:

```
obj = 48;
times = [{2,4,10},{3,6,7,8},{1,5,9}];
```

3.2 Dicas de Implementação

Este problema é uma combinação do multi-mochila com o de atribuição $m \times n$. O problema do cabo-de-guerra dá uma idéia de como generalizar o problema.

3.3 Quanto ao *search*

Opções do *search* do Minizinc para busca com **inteiros** são resumidos em:

```
.....
%%solve :: int_search([LIST OF VARS], SELECT_VAR, DOMAIN_VAR, complete) minimize CO
%% SELECT_VAR_INT={input_order, anti_first_fail, first_fail, smallest, largest, max_r
%% DOMAIN_VAR_INT={indomain, indomain_split, indomain_random, indomain_median, indoma
%% DETAILS in http://www.minizinc.org/g12_www/zinc/doc-2.0/html/zinc-manual.h
.....
```

Troque os parâmetros da tabela.