Bebras PT

Edição 2025

Categoria
Seniores (11º e 12º ano de escolaridade)

Tempo
45 minutos

Resolve tantos problemas quanto possível em 45 minutos.

Não é esperado que consigas resolver todos!

Responde apenas na folha de respostas.

É uma folha única, à parte, que deverás identificar com o teu nome.

Os enunciados e folhas de rascunho devem ser obrigatoriamente recolhidos no final da prova.

TREETREE2 FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO Deloitte. FUNDAÇÃO CALOUSTE GULBENKIAN

O **Bebras** é uma iniciativa internacional destinada a promover o pensamento computacional e a Informática (Ciência de Computadores). Foi desenhado para motivar alunos de todas as idades mesmo que não tenham experiência prévia.

Esta iniciativa começou em 2004 na Lituânia e todos os anos participam mais de 3 milhões de alunos de todo o mundo. O seu nome original vem dessa origem - "bebras" significa "castor" em lituano. A comunidade internacional adotou esse nome, porque os castores buscam a perfeição no seu dia-a-dia e são conhecidos por serem muito trabalhadores e inteligentes.

O que é o Pensamento Computacional?

O pensamento computacional é um conjunto de técnicas de resolução de problemas que envolve a maneira de expressar um problema e a sua solução de modo a que um computador (seja um humano ou máquina) a possa executar. É muito mais do que simplesmente saber programar. O desafio do Bebras promove precisamente este tipo de habilidades e conceitos como a capacidade de partir um problema complexo em problemas mais simples, o desenho de algoritmos, o reconhecimento de padrões ou a capacidade de generalizar e abstrair.

Organização Portuguesa

O Bebras começou em **Portugal** em 2019 e no ano passado contou com a participação de 136 232 alunos, de 849 escolas de Portugal, Angola, Moçambique, Cabo Verde e Timor-Leste.

É organizado por uma equipa de pessoas ligadas à Educação e à Ciência de Computadores da **TreeTree2** e do Departamento de Ciência de Computadores da Faculdade de Ciências da Universidade do Porto (**DCC/FCUP**)

Estrutura da Prova

Existe apenas uma fase a nível nacional, a qual é constituída por uma prova individual com 12 questões de três níveis de dificuldade diferentes, cuja pontuação é da seguinte forma:

Dificuldade	Correto	Incorreto	Não respondido
fácil	+6 pontos	-2 pontos	0 pontos
média	+9 pontos	-3 pontos	0 pontos
difícil	+12 pontos	-4 pontos	0 pontos

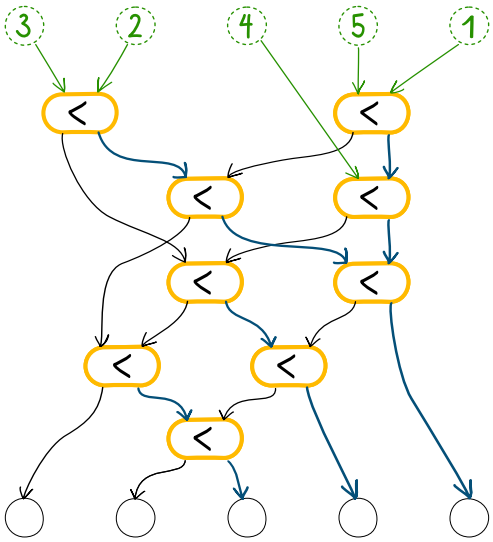
Sobre os Problemas

Os problemas da edição portuguesa foram escolhidos, traduzidos e adaptados pela organização portuguesa. Para a deste ano foram usados problemas com autores originários dos seguintes países:

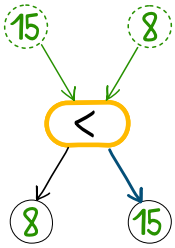
Alemanha	Argentina	Brasil	Bulgária	Canadá
Chéquia	Coreia do Sul	Costa Rica	Eslovénia	Eslováquia
Indonésia	Japão	Lituânia	Malásia	México
Montenegro	Paquistão	Portugal	Reino Unido	Suíça

1. Uma Máquina Especial

Os castores têm uma máquina especial que recebe cinco números nas entradas do topo (por exemplo 3, 2, 4, 5 e 1). Através de setas e interruptores, os números percorrem a máquina até chegarem às saídas na parte inferior.



Cada um dos nove interruptores < compara os dois números que recebe e encaminha o número mais pequeno para a esquerda, e o número maior para a direita, como representado no exemplo abaixo.



Pergunta

Que tarefa é executada por esta máquina?

Respostas possíveis

(A) Ordena os números pela ordem inversa da entrada (resultado: 1, 5, 4, 2, 3)

(B) Ordena os números por ordem crescente (resultado: 1, 2, 3, 4, 5)

(C) Ordena os números por ordem decrescente (resultado: 5, 4, 3, 2, 1)

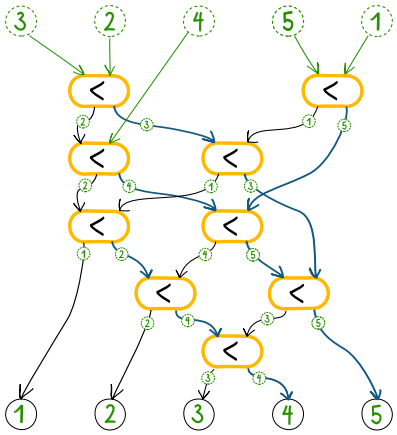
(D) Mantém a ordem de entrada (resultado: 3, 2, 4, 5, 1)

1. Uma Máquina Especial (Resolução)

Solução

(B)

Resolução



Isto é Pensamento Computacional!

Em informática, a máquina especial desta tarefa Bebras é conhecida como "rede de ordenação" (*sorting network*). É construída a partir de vários componentes idênticos e muito simples: os comparadores. Cada comparador recebe dois valores numéricos em duas linhas e compara-os, encaminhando o valor mais pequeno para a linha da esquerda e o valor maior para a linha da direita.

Combinando um número suficiente de comparadores, qualquer sequência de números pode ser ordenada por valor! Para ordenar cinco números, são necessários pelo menos nove comparadores. Para seis números, são necessários doze comparadores.

As redes de ordenação têm relevância prática pelas seguintes razões:

1. Em geral, uma ordenação rápida e eficiente facilita a pesquisa, por exemplo, em listas de resultados de motores de busca.
2. O algoritmo de ordenação é adequado para execução paralela, podendo ser executado de forma muito rápida.
3. Estas redes são muito adequadas para implementação em hardware, por exemplo em GPUs (*Graphics Processing Units*), porque os comparadores podem ser construídos como dispositivos eletrónicos muito simples e, portanto, baratos.

Algumas desvantagens incluem:

1. As redes de ordenação são concebidas apenas para entradas de comprimento fixo.
2. O número de comparadores necessários aumenta rapidamente com o comprimento da entrada.

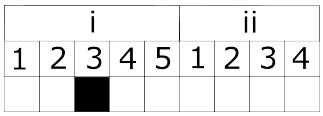
2. Mensagem Oculta

Um grupo de castores usa um código secreto que substitui cada letra pelo número que corresponde à sua posição no alfabeto. A tabela abaixo mostra o alfabeto completo com o número correspondente.

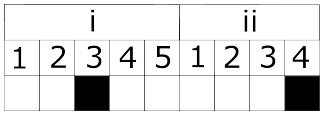
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Cada letra é depois codificada numa linha de 9 quadrados, dividida em duas partes (i e ii). Os passos para codificar cada letra são os seguintes:

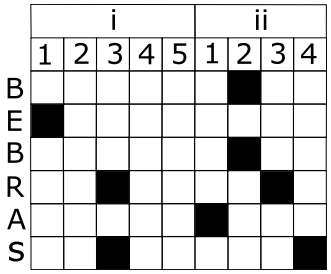
1. Encontrar o número da letra no alfabeto (por exemplo, o número do S é 19).
2. Dividir este número por 5, arredondando para baixo, e pintar de preto a caixa desta posição na secção (i). Por exemplo, 19/5 arredondado para baixo dá 3.



3. Calcular o resto do número da letra dividido por 5 e pintar de preto a caixa desta posição na secção (ii). Por exemplo, 19/5 tem resto 4.



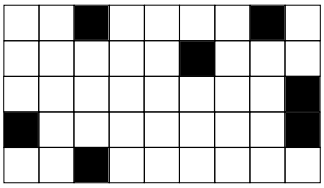
Assim, quando um castor quer enviar a palavra "BEBRAS", ela será codificada como:



Os castores vão transformar a tabela de letras codificadas numa imagem, mas não vão usar a parte de cima onde estão os títulos.

Pergunta

Qual a mensagem codificada na seguinte grelha?



Resposta

Escreve a mensagem codificada na folha de respostas.

2. Mensagem Oculta (Resolução)

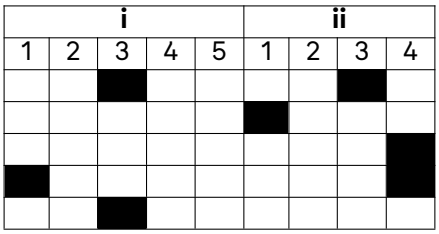
Solução

RADIO

Resolução

Para encontrar as letras escondidas, temos de encontrar primeiro o número codificado pela grelha para cada linha através da seguinte fórmula:

número da letra = posição da caixa preta em (i) $\times 5$ + posição da caixa preta em (ii)



número = $(3 \times 5) + 3 = 18$ (R)
número = $(0 \times 5) + 1 = 1$ (A)
número = $(0 \times 5) + 4 = 4$ (D)
número = $(1 \times 5) + 4 = 9$ (I)
número = $(3 \times 5) + 0 = 15$ (O)

Isto é Pensamento Computacional!

Esta tarefa introduz o conceito de **codificação**, ou seja, a conversão de dados de um formato para outro. Neste caso, as letras do alfabeto são primeiro transformadas em números e, depois, esses números são representados como um padrão visual utilizando quadrados. O desafio consiste em descodificar essa representação visual e reconstruir a mensagem original — uma sequência de letras.

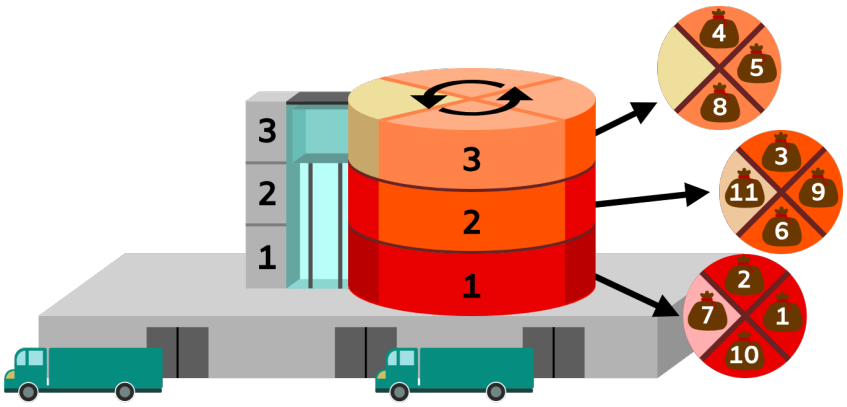
Na ciência de computadores, os dados nem sempre são escritos como texto simples ou números. Por vezes, são transformados num formato visual que pode ser lido e descodificado por um computador. Dois exemplos do dia a dia são os **códigos de barras** e os **códigos QR**.

Nesta tarefa, a divisão com arredondamento e resto é usada como método para codificar e descodificar dados. Esta técnica desempenha um papel importante em muitas áreas da ciência de computadores e dos sistemas digitais. Por exemplo, sistemas de encriptação como o **RSA** utilizam operações de módulo (restos) para transformar de forma segura dados em mensagens codificadas e depois recuperá-las.

3. Fluxo de Presentes

És um(a) programador(a) a trabalhar na "GiftFlow", um armazém de entrega de presentes. O teu trabalho é escrever instruções para um robô que move presentes do armazém para a zona de descarga.

O armazém tem 4 níveis: zona de descarga, Nível 1, Nível 2 e Nível 3. Cada nível de armazenamento é uma plataforma rotativa dividida em 4 setores. Os presentes são guardados em sacos nestes setores, havendo no máximo um saco por setor. A zona de descarga encontra-se no rés-do-chão.



Os comandos que podes dar ao robô são os seguintes:

- MU: subir um andar no elevador
- MD: descer um andar no elevador
- R: rodar a plataforma do andar em que o robô se encontra um setor no sentido contrário ao dos ponteiros do relógio
- L: carregar o presente do setor a que o robô tem acesso para o elevador
- U: descarregar o presente que está há mais tempo no elevador

Exemplo

Imagina que o robô começa no andar 2 e quer descarregar os presentes 8 e 9 no rés-do-chão. Uma sequência de comandos que permite executar esta tarefa é a seguinte: **MU, R, R, R, L, MD, R, R, L, MD, MD, U, U**.

Esta lista de comandos move o robô um andar para cima (MU), roda esse nível três vezes (R, R, R), carrega o saco 8 (L), desce um nível (MD), roda duas vezes (R, R), carrega o saco 9 (L) e depois leva os dois sacos para baixo para serem descarregados (MD, MD, U, U).

Pergunta

O elevador está atualmente no nível 1 e está vazio. Os sacos estão dispostos conforme mostrado na imagem. O robô executa a seguinte sequência: **R, R, L, MU, MU, R, R, MD, R, L, MD, MD, U, U**

Que sacos são descarregados na área de descarga?

Respostas possíveis

(A) 3, 5

(B) 1, 5

(C) 1, 3, 5

(D) 1, 3

3. Fluxo de Presentes (Resolução)

Solução

(D)

Resolução

Temos apenas dois comandos de carga (L) e dois comandos de descarga (U), por isso carregamos/descarregamos apenas dois sacos. Assim, a opção c) não é correta. As opções a) 3, 5 e b) 1, 5 não são correctas porque o robô não carregou o saco 5. Se analisarmos os comandos em detalhe, temos:

- R, R, L — carrega o saco 1
- MU, MU, R, R, MD — move para o nível 2
- R, L — carrega o saco 3
- MD, MD, U, U — desce e descarrega os dois sacos, 1 e 3

Isto é Pensamento Computacional!

Decomposição: A tarefa global (mover presentes entre os níveis do armazém) é decomposta em sub-tarefas mais pequenas e geríveis:

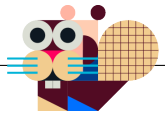
- Mover o elevador verticalmente (MU, MD).
- Rodar o nível do armazém (R).
- Carregar um presente (L).
- Descarregar um presente (U).

Pensamento algorítmico:

- *Instruções passo a passo:* As ações do robô são definidas como uma sequência precisa de passos (a sequência de comandos). Isto é um aspeto central do pensamento algorítmico.
- *Desenvolver uma estratégia:* Para mover os presentes, é necessário conceber uma estratégia (um algoritmo) que envolva:
  - Determinar as localizações iniciais e finais dos presentes.
  - Calcular os movimentos verticais necessários do elevador.
  - Calcular as rotações necessárias.
  - Inserir os comandos de carregar e descarregar nos pontos corretos.
- *Exemplo como algoritmo:* O exemplo dado (MU, R, R, R, L, MD, R, R, L, MD, MD, U, U) é um algoritmo específico para mover os presentes 8 e 9.
- *Otimização:* O pensamento algorítmico também envolve considerar a eficiência. Poderia o movimento dos presentes ser realizado com menos passos?

Raciocínio espacial:

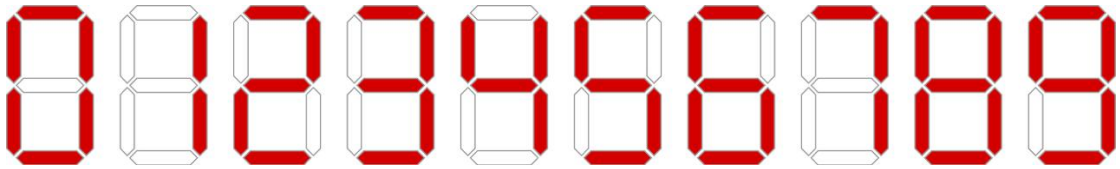
- *Compreensão da disposição do armazém:* O robô precisa de compreender a disposição espacial dos níveis e setores. Isto envolve visualizar as posições relativas das diferentes localizações.



- *Calcular rotações:* Determinar o número de rotações necessárias para alcançar um setor específico requer raciocínio espacial. O robô tem de acompanhar mentalmente o seu setor atual e o setor alvo.
- *Visualizar movimentos verticais:* O robô precisa de perceber a relação vertical entre os níveis para executar corretamente os comandos do elevador.
- *Gestão do elevador:* O robô precisa de acompanhar a ordem dos presentes dentro do elevador e visualizar qual presente será descarregado a seguir.
- *Posições relativas:* O comando « rodar o andar em que o robô se encontra um setor no sentido contrário ao dos ponteiros do relógio» exige raciocínio espacial, pois o robô deve visualizar o movimento dos presentes dentro do elevador.

4. Segmento Desgastado

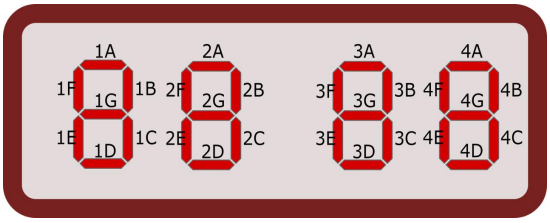
Um relógio digital padrão, que muda a cada minuto, mostra as horas usando quatro dígitos. Cada dígito contém 7 segmentos luminosos. Utilizando esses segmentos, cada dígito consegue representar os números de 0 a 9:



Os segmentos desgastam-se cada vez que são ativados (ou seja, quando passam do estado desligado para ligado). O segmento que for ativado mais vezes será o primeiro a precisar de ser substituído.

Pergunta

Qual dos 28 segmentos do relógio digital é o primeiro a precisar de ser substituído?



Resposta

Na folha de respostas, indica o código que representa o segmento, seguindo o esquema representado acima.

4. Segmento Desgastado (Resolução)

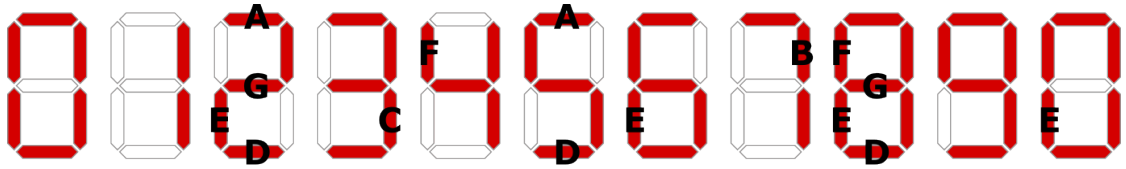
Solução

4E

Resolução

Cada dígito no mostrador representa um período de tempo diferente: dezenas de horas, horas, dezenas de minutos e minutos. O dígito mais à direita — ou seja, o dígito 4 — é o que muda com mais frequência, uma vez que representa os minutos individuais. Antes que qualquer outro dígito mude, todos os segmentos do dígito mais à direita já terão mudado várias vezes. O segmento mais desgastado encontra-se, portanto, no dígito 4, e o seu código de segmento é E.

Devemos então analisar todos os 7 segmentos do dígito mais à direita para determinar quantas vezes cada um é ativado (ou seja, ligado) quando o dígito representado passa de 0 a 0 novamente:



A imagem mostra essa verificação. Um segmento ativado (acabado de acender) é aquele que está iluminado no dígito atual, mas não estava iluminado no dígito imediatamente anterior. Cada ativação de um segmento é assinalada com a letra correspondente a esse segmento. O número total de ativações de um segmento é o número de vezes que a sua letra aparece na imagem.

Segmento	Dígito que causa a ativação do fragmento	Número de vezes que o segmento é ativado
A	2, 5	2
B	7	1
C	3	1
D	2, 5, 8	3
E	2, 6, 8, 0	4
F	4, 8	2
G	2, 8	2

A tabela resume esta verificação:

- A coluna central mostra os dígitos em que o segmento foi ativado.
- A coluna da direita contabiliza essas ativações.

O segmento que muda mais vezes é o E no dígito 4, e o seu código é 4E.

Isto é Pensamento Computacional!

Este problema exige que os alunos usem o raciocínio lógico para perceber que o dígito mais à direita de um relógio digital padrão é o que muda com mais frequência. Assim, o problema reduz-se a analisar o padrão de mudanças dos segmentos no dígito mais à direita.

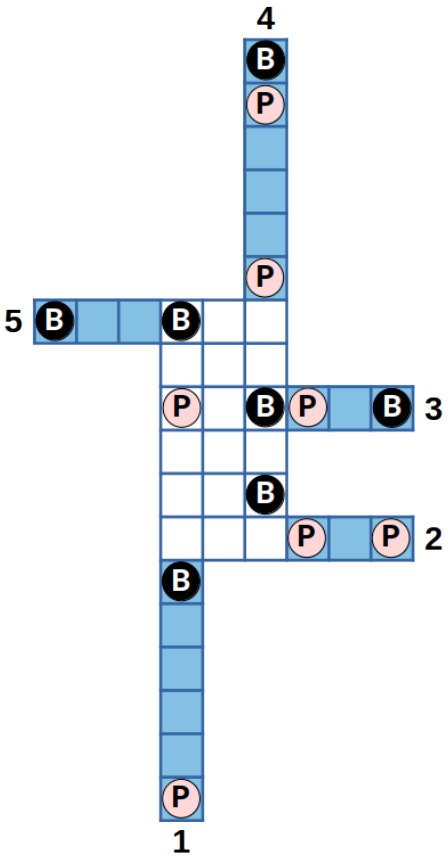
De seguida, os alunos precisam de analisar o número de mudanças em cada segmento do visor de sete segmentos mais à direita durante um ciclo completo de mudança. Esta mudança é representada pela transição dos segmentos do visor de desligado para ligado. Este processo requer a capacidade de reconhecer padrões.

5. Dobragem Perfeita

A figura abaixo representa uma grelha 3 por 6 com algumas linhas e colunas prolongadas (numeradas e a azul). Na grelha e nos prolongamentos, alguns quadrados contêm pinos (P) vermelhos, outros contêm buracos (B) pretos e outros estão limpos. O Lucas quer dobrar as 5 extensões azuis da figura de forma a obter uma dobragem perfeita, ou seja, uma grelha 3 por 6 sem extensões. Durante a dobragem, o Lucas deve considerar as seguintes regras:

- Ao sobrepor um pino sobre um buraco, ou um buraco sobre um pino, o quadrado resultante fica limpo;
- Ao sobrepor um buraco sobre outro buraco, o buraco mantém-se;
- Quando um quadrado limpo é sobreposto a um buraco, ou um buraco a um quadrado limpo, o buraco prevalece;
- É impossível sobrepor um pino a outro pino;
- É impossível sobrepor um quadrado limpo a um pino, ou um pino a um quadrado limpo;

Para além disso, uma dobragem só é considerada válida se todos os quadrados da grelha ficarem limpos no final do processo.



Pergunta

Por quantas ordens diferentes podemos dobrar os segmentos para obter uma dobragem perfeita?

Resposta

Escreve o número de possibilidades na folha de resposta.

5. Dobragem Perfeita (Resolução)

Solução

6

Resolução

Para chegar à lista de possibilidades, é possível listar todas as ordens de dobragem e excluir aquelas que não funcionam. No entanto, isto corresponde a considerar 5!=120 possibilidades, pelo que idealmente o problema deve ser resolvido com uma análise mais global:

- A aba 2 tem de ser depois da 1 e da 4;
- A aba 3 tem de ser antes da 1;
- A aba 5 tem de ser antes da 4.

Isto significa que a aba 2 tem de ser a última. Assim, o problema fica reduzido a escolher a posição de 2 dos números (por exemplo, 3 e 1, uma vez que as posições do 5 e do 4 ficam autodeterminadas ao colocar estes dois números) das quatro posições disponíveis, o que resulta em 6 ordens possíveis:

- 3 1 5 4 2
- 3 5 1 4 2
- 3 5 4 1 2
- 5 3 1 4 2
- 5 3 4 1 2
- 5 4 3 1 2

Isto é Pensamento Computacional!

Resolver esta tarefa exige a aplicação de várias ferramentas mentais do pensamento computacional, tais como:

- **Decomposição:** dividir um problema complexo em partes mais simples e geríveis — analisar a estrutura da grelha e a localização dos pinos, buracos e espaços limpos, identificar possíveis estratégias de dobragem e avaliar se a dobragem conduz a uma solução válida;
- **Desenvolvimento de algoritmos:** o problema é um caso de ordenação com restrições;
- **Abstração:** podemos representar a grelha como um array de valores (por exemplo, 1 para pinos, -1 para buracos, 0 para limpo) e, em vez de nos focarmos nos detalhes gráficos, pensar em como modificar os valores da matriz ao realizar as dobras.

6. Raios Cósmicos

A Marta está a contar a uma amiga uma história estranha sobre a sua conta bancária. Um dia, a Marta consultou o saldo no telemóvel e viu que tinha 8095 euros. De repente, o número mudou para 7071 euros. Como não tinha feito nenhuma transação nem pagamento, a Marta ficou confusa e enviou uma mensagem ao banco a perguntar o que tinha acontecido.

Os funcionários do banco analisaram a situação e concluíram que não havia erro no software nem tinha havido nenhum ciberataque. Acabaram por perceber que tinha havido uma alteração aleatória no valor de um bit na memória, provocada por radiação cósmica.

Nos sistemas informáticos, os números são armazenados em binário. O sistema de numeração binária usa apenas dois dígitos: 0 e 1. Os valores das posições aumentam como múltiplos de dois à medida que se avança no número da direita para a esquerda (do dígito menos significativo para o mais significativo).
A tabela abaixo mostra como ler o número de 8 bits "00100011":

128	64	32	16	8	4	2	1
0	**0**	**1**	**0**	**0**	**0**	**1**	**1**

Em formato decimal, este número é 32+2+1=35.

Na imagem seguinte um único bit mudou, nomeadamente o **8º bit** (porque contamos as posições da direita para a esquerda) e ficamos com o valor 128+32+2+1=163.

128	64	32	16	8	4	2	1
1	**0**	**1**	**0**	**0**	**0**	**1**	**1**

Como podes ver, a alteração de um único bit pode mudar substancialmente o valor do número. Um saldo bancário teria de ser armazenado usando um grande número de bits, já que o saldo pode ser muito elevado.

Pergunta

Qual foi o bit do saldo da Marta que foi alterado pela radiação cósmica?

Respostas possíveis

(A) 11º bit de 1 para 0

(B) 12º bit de 1 para 0

(C) 11º bit de 0 para 1

(D) 12º bit de 0 para 1

6. Raios Cósmicos (Resolução)

Solução

(A)

Resolução

Primeiro, descobrimos quanto mudou o saldo da conta, subtraindo os valores:

$$8095 - 7071 = 1024$$

Pelas tabelas, podemos ver que o valor de posição de cada bit duplica à medida que avançamos da direita para a esquerda. Para chegar ao valor de 1024 precisamos de encontrar o 11.º bit.
Podemos então verificar os números e ver que, na forma binária, eles diferem apenas num dígito, no 11.º bit:

$$8095\ (\text{base } 10) = 1\ 1111\ 1001\ 1111\ (\text{base } 2)\ \text{vs.}\ 7071\ (\text{base } 10) = 1\ 1011\ 1001\ 1111\ (\text{base } 2)$$

Isto é Pensamento Computacional!

Os sistemas informáticos armazenam números em forma binária. Estes números são guardados na memória do computador, que pode estar sujeita a erros e avarias. Normalmente, a origem é elétrica, e os valores 0 ou 1 são armazenados carregando ou descarregando um componente.

É um facto que a radiação cósmica, composta por partículas carregadas rápidas, também pode atingir a superfície da Terra e, portanto, atravessar a memória de um computador. Quando atinge uma posição da memória que guarda o valor de um bit de informação, pode carregá-lo ou descarregá-lo. A probabilidade é muito pequena, mas existe.

As naves espaciais estão em ainda maior perigo, porque a radiação cósmica é muito mais forte no espaço. É por isso que usam três blocos de memória para armazenar os mesmos valores. Durante o funcionamento, verificam constantemente se os valores são diferentes entre si e, caso sejam, descobrem qual dos três valores é diferente dos outros e corrigem-no.

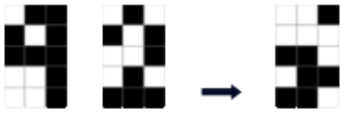
Existem vários métodos específicos de deteção e correção de erros (como os códigos de Hamming, os códigos de Reed-Solomon, etc.), com diferentes níveis de complexidade e capacidades de correção de erros, para lidar com erros de dados.

As componentes de Pensamento Computacional avaliadas neste problema incluem:

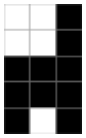
- **Abstração**: reconhecer o que é importante numa solução e concentrar-se apenas nessa componente
- **Reconhecimento de padrões**: padrões no valor de posição dos números binários

7. Quadriculas em Código

A Gema está aborrecida durante uma aula e começa a preencher quadrados na sua folha quadriculada para desenhar números. Ela inventa uma forma especial de representar o seu número favorito de dois dígitos, o 42. A partir das duas representações dos algarismos "4" e "2" numa grelha 3 × 5, mostradas abaixo, ela cria uma nova grelha em que cada quadrado é pintado de preto se, e só se, exatamente um dos dois quadrados correspondentes nas imagens do "4" e do "2" estiver a preto.



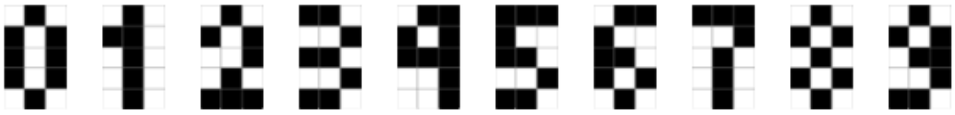
A Gema utilizou o mesmo método para representar um outro número de 2 dígitos, e obteve a seguinte figura:



Pergunta

Qual foi o número de 2 dígitos escolhido pela Gema para obter esta representação?

Se houver mais do que uma resposta correta, indica apenas uma. As imagens iniciais correspondentes a cada dígito são dadas a seguir.



Resposta

Escreve dois dígitos na folha de respostas.

7. Quadriculas em Código (Resolução)

Solução

Há duas respostas corretas possíveis, 26 e 62.

Resolução

Combinar estes dois dígitos seguindo a regra definida pela Gema resulta no padrão indicado:

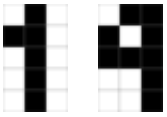
Note-se que o método de combinação é simétrico em relação aos dois dígitos do número selecionado, o que significa que dois números com o mesmo par de dígitos em ordem inversa (por exemplo, 26 e 62) dão origem à mesma representação.

Para resolver esta tarefa, temos de encontrar formas de determinar os dois algarismos corretos evitando procurar por todas as combinações possíveis, uma vez que há 100 no total, de "00" a "99". Como normalmente omitimos o zero à esquerda em números de um só algarismo (escrevemos "7" e não "07"), isto reduz o número de combinações para 90, ou até para 45 se considerarmos a simetria mencionada acima. Ainda assim, este número é demasiado elevado para verificar exaustivamente cada possibilidade.

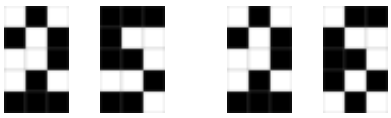
A primeira coisa a compreender é que a operação de combinação produz um pixel preto numa determinada posição sempre que esse pixel é diferente nos dois algarismos de origem. Isto resulta do facto de que obtemos um pixel preto no resultado sempre que há apenas um pixel preto ao considerar as imagens de origem. Se olharmos para um único pixel e listarmos todas as possibilidades, obtemos esta tabela de correspondência:
[inserir tabela]
Ao perceber isto, podemos de imediato excluir todos os números formados pelo mesmo algarismo repetido, como "22" ou "99", já que todos os pixels seriam iguais entre os dois algarismos e isso resultaria numa imagem combinada completamente branca.

Uma forma de abordar o resto da procura pela solução é notar que a coluna mais à direita da imagem combinada é toda preta, o que indica que os dois algarismos de origem têm de ter todos os pixels diferentes nessa coluna.

Um par de algarismos óbvio que satisfaz esta condição é "1"–"4" — a última coluna está toda vazia no "1" e totalmente preenchida no "4":



Infelizmente, ao observarmos, por exemplo, a segunda linha ou a linha inferior, podemos excluir esse par, pois o padrão resultante não corresponde à imagem combinada. Os dois outros pares, um pouco menos óbvios, em que todos os pixels são diferentes na última coluna, são "2"–"5" e "2"–"6":



Ao inspecionarmos os pixels restantes da imagem combinada (por exemplo, o primeiro pixel na linha superior), podemos excluir o par "2"–"5". Ficamos assim apenas com o par "2"–"6", que resulta na representação desejada. Considerando a simetria mencionada acima, concluímos que tanto 26 como 62 são respostas possíveis.

Isto é Pensamento Computacional!

A operação central utilizada na criação da imagem combinada é, na verdade, a operação booleana OU-exclusivo (**XOR**). Uma operação booleana atua sobre valores verdadeiro/falso (chamados valores booleanos) e produz outro valor booleano. Esta é a tabela de verdade da operação booleana XOR, onde Z é o resultado:
[inserir tabela]

Nota-se a semelhança com a tabela de correspondência de pixels mostrada acima, se substituirmos os 0 por quadrados vazios e os 1 por quadrados preenchidos a preto.

Outras operações booleanas incluem **E** e **OU**. Estas operações são amplamente usadas no desenho de sistemas de hardware sob a forma de portas lógicas. São blocos de construção que recebem dois sinais como entrada e "calculam" a saída de acordo com a respetiva tabela de verdade.

Em criptografia, o XOR desempenha um papel central em esquemas de encriptação simétrica devido à sua natureza reversível: aplicar a mesma operação XOR duas vezes com a mesma chave resulta no valor original (neste exemplo, combinar a imagem combinada com uma das imagens dos algarismos dará como resultado... a outra imagem do algarismo!). Esta propriedade torna o XOR útil tanto para codificar como para descodificar dados. O XOR é também amplamente utilizado em técnicas de deteção e correção de erros, como checksums e códigos de redundância cíclica (CRC), onde esta operação ajuda a identificar discrepâncias entre os dados transmitidos e recebidos.

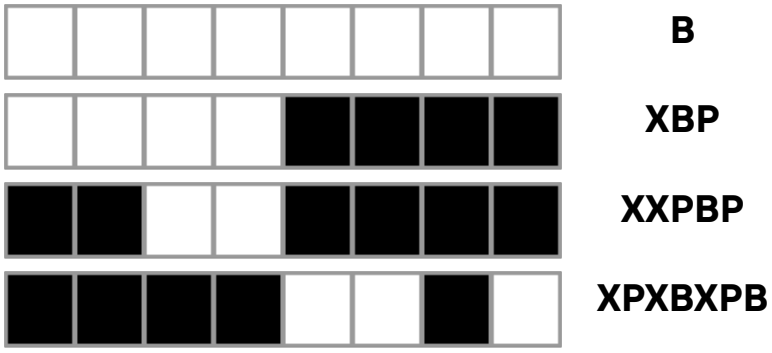
Esta tarefa treina sobretudo o **reconhecimento de padrões**, ao verificar pixels nas imagens de origem e na imagem combinada.

8. Padrão em Linha

A Sara está a jogar um jogo. Ela tem uma sequência de quadrados que podem estar pintados de preto ou branco, e quer representá-los de uma forma específica, segundo as seguintes regras:

- Se todos os quadrados da sequência atual forem brancos, escreve apenas 'B'
- Se todos os quadrados da sequência atual forem pretos, escreve apenas 'P'
- Caso contrário, escreve 'X'
  - seguido pelo resultado da mesma regra aplicado à metade esquerda da sequência atual...
  - ... e depois pelo resultado da mesma regra aplicado à metade direita da sequência atual

Eis alguns exemplos de como a regra funciona para uma sequência de 8 quadrados:



Pergunta

Como deve a Sara representar a sequência seguinte?

Respostas possíveis

(A) XXPBPXBXBP

(B) XXXPBPXBXBP

(C) XXXBPXBBXXPPXPB

(D) XBPBXPPB

(E) XXXBPBXPXPB

(F) XXBPBXPXPB

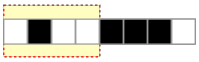
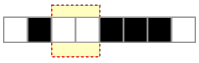
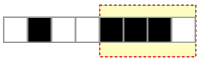
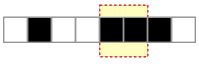
8. Padrão em Linha (Resolução)

Solução

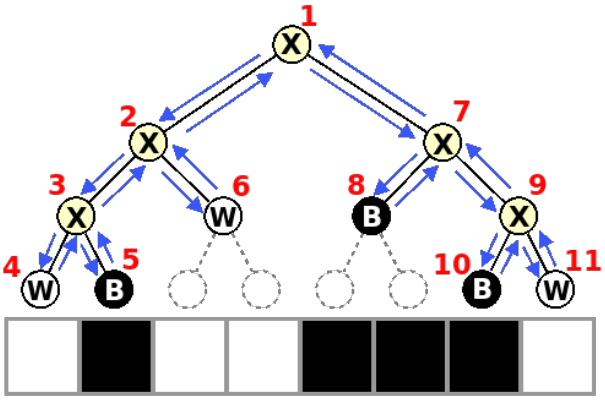
(E)

Resolução

Podemos construir a resposta correta letra a letra da seguinte forma:

-  X - Os 8 quadrados não são todos da mesma cor, por isso começamos com um X
-  XX - A metade esquerda não é toda da mesma cor, por isso precisamos de outro X
-  XXX - A metade esquerda da metade esquerda não é toda da mesma cor
-  XXXB - o quadrado que sobra do lado esquerdo é branco
-  XXXBP - o quadrado que sobra do lado direito é preto
-  XXXBPB - A metade direita da metade esquerda é branca
-  XXXBPBX - A metade direita original não é toda da mesma cor
-  XXXBPBXP - A metade esquerda da metade direita é preta
-  XXXBPBXPX - A metade direita da metade direita não é toda da mesma cor
-  XXXBPBXPXP - o quadrado que sobra do lado esquerdo é preto
-  XXXBPBXPXPB - o quadrado que sobra do lado direito é branco

Outra forma de visualizar a resolução deste problema é dada pela imagem seguinte, com as setas azuis a indicar como “nos deslocamos” e os números vermelhos a indicar em que ordem “escrevemos” a representação:



Como podemos construir esta árvore? Primeiro escrevemos uma sequência de letras para cada quadrado: BPBBPPPB. Assim obtemos o nível mais baixo da árvore, as folhas. Depois subimos para os nós superiores. Se ambos os filhos forem iguais (apenas B ou apenas P), combinamo-los numa única letra e eliminamos os dois filhos. Isso torna-se numa nova folha da árvore. Caso contrário, escrevemos X no nó.

Como construir a sequência correta a partir da árvore? Começamos pelo nó superior. Percorremos a árvore indo sempre para a esquerda em cada interseção primeiro. Em cada nó por onde passamos, escrevemos a letra que aparece no nó na primeira vez que o visitamos. Quando chegamos a uma folha, regressamos ao nó anterior e percorremos o ramo direito. Se ambos os ramos já tiverem sido visitados, regressamos ao nó anterior.

A opção (B) está incorreta porque dá a linha "complementar" (preto quando o quadrado é branco e vice-versa). A opção (C) está incorreta porque tem secções de dois quadrados iguais que não são reduzidas a uma única letra e estão em vez disso divididas em duas letras iguais. As opções (A), (D) e (F) estão incorretas, entre outras razões, porque não começam com três X como é necessário para representar os primeiros quadrados.

Isto é Pensamento Computacional!

Nesta tarefa, é necessária abstração para compreender a definição recursiva apresentada no corpo do texto. Podemos então abordar o problema passo a passo, decompondo-o em problemas menores e simulando/avaliando cada passo para obter a representação original dos dados.

9. Tecla Mágica

O Tiago tem um rato de computador especial que tem uma tecla mágica! O rato mantém um contador, que é incrementado em 1 cada vez que esta tecla mágica é utilizada.

Para além disso, cada vez que esta tecla é pressionada, todos os números inteiros entre o valor atual do contador e 1 são escritos num ficheiro especial. Como o contador começa com o valor 1, depois de o Tiago carregar na tecla 5 vezes, este ficheiro especial contém a seguinte sequência: **1, 2, 1, 3, 2, 1, 4, 3, 2, 1, 5, 4, 3, 2, 1**

Nº de cliques	Conteúdo do ficheiro
1	**1**
2	1, **2, 1**
3	1, 2, 1, **3, 2, 1**
4	1, 2, 1, 3, 2, 1, **4, 3, 2, 1**
5	1, 2, 1, 3, 2, 1, 4, 3, 2, 1, **5, 4, 3, 2, 1**

Pergunta

O Tiago está fascinado com este fenômeno e carregou na tecla mágica várias vezes, gerando um ficheiro com uma sequência muito comprida.

Qual o número que se encontra na 127ª posição da sequência?

Resposta

Escreve um número inteiro (entre 1 e 16) na folha de respostas.

9. Tecla Mágica (Resolução)

Solução

10

Resolução

O tamanho do ficheiro (em termos da contagem de números nele escritos) é, após carregar na tecla mágica algumas vezes: 1, 3, 6, 10, 15, 21, 28... De forma geral, se **n** é o número vezes que se carregou na tecla, o tamanho do ficheiro será $\mathbf{T(n) = n\,(n+1)\,/\,2}$.

Para obter a resposta, podemos recorrer a uma simulação. Podemos tentar responder à seguinte questão: qual é o menor **n** tal que $\mathbf{T(n) \geq 127}$? Claramente, qualquer n que produza um valor mais pequeno não pode ser a resposta, pois isso significaria que menos de 127 números foram escritos no ficheiro. Assim, aumentamos progressivamente n até encontrarmos o primeiro número que satisfaça a condição. Verificamos que $\mathbf{T(15) = 120}$, que é demasiado pequeno, mas $\mathbf{T(16) = 136}$.

Quando carregamos no botão pela 16.ª vez, os números escritos serão **[16, 15, 14, 13, 12, 11, 10, ..., 1]**. Sabemos que o 16 é o 121.º número escrito. Então, basta contar para baixo a partir do 16 até chegarmos ao 127.º número, que será **10**.

Isto é Pensamento Computacional!

Para resolver esta tarefa, é essencial aplicar conhecimentos de reconhecimento de padrões. Reconhecer a sequência pela qual o tamanho do ficheiro aumenta é semelhante a identificar padrões que podem ajudar a produzir melhores algoritmos.

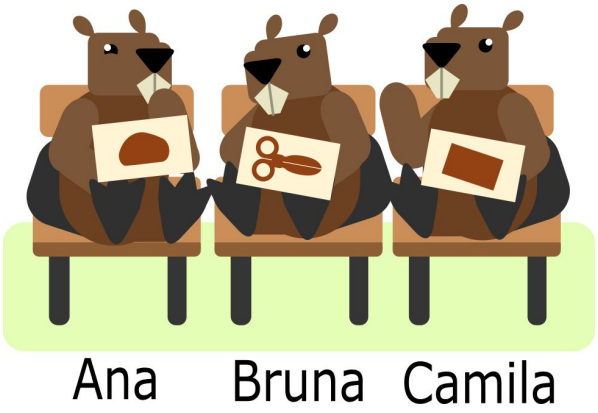
A capacidade de decomposição também é útil nesta tarefa. O estudante pode decompor o espaço do problema em duas partes: todos os números que podem ser ignorados e o conjunto de números relevantes onde a resposta deve ser encontrada.

Estas competências promovem uma compreensão profunda do pensamento computacional através de cenários simples.

10. Pedra, Papel, Tesoura, Troca

A Ana, a Bruna e a Camila inventaram uma nova versão do jogo pedra, papel, tesoura. Na versão original deste jogo, a pedra ganha à tesoura, a tesoura ganha ao papel e o papel ganha à pedra.

Na versão modificada, a Ana, a Bruna e a Camila sentam-se em cadeiras e seguram um cartão com o objeto que vão utilizar, de forma a que todas possam ver a sua jogada. Antes de determinar quem ganha a quem, as jogadoras têm de decidir entre si quantas trocas vão efetuar. Uma troca é uma operação realizada entre duas jogadoras apenas, que trocam os seus cartões. Depois de decidir quantas trocas vão realizar, as três amigas têm de escolher que jogadoras estarão envolvidas em cada uma delas.



Pergunta

O único objetivo da Bruna nesta ronda é ganhar à Camila. Que estratégia garante que a Bruna tem sucesso na sua tarefa?

Respostas possíveis

(A) A Bruna só tem de garantir um número ímpar de trocas com a Camila

(B) Independentemente do número de trocas realizadas, a Bruna nunca deve trocar de cartão com a Camila

(C) Independentemente do número de trocas realizadas, a Bruna deve trocar sempre de cartão com a Camila

(D) A Bruna só tem de garantir que há um número par de trocas, independentemente das jogadoras nelas envolvidas

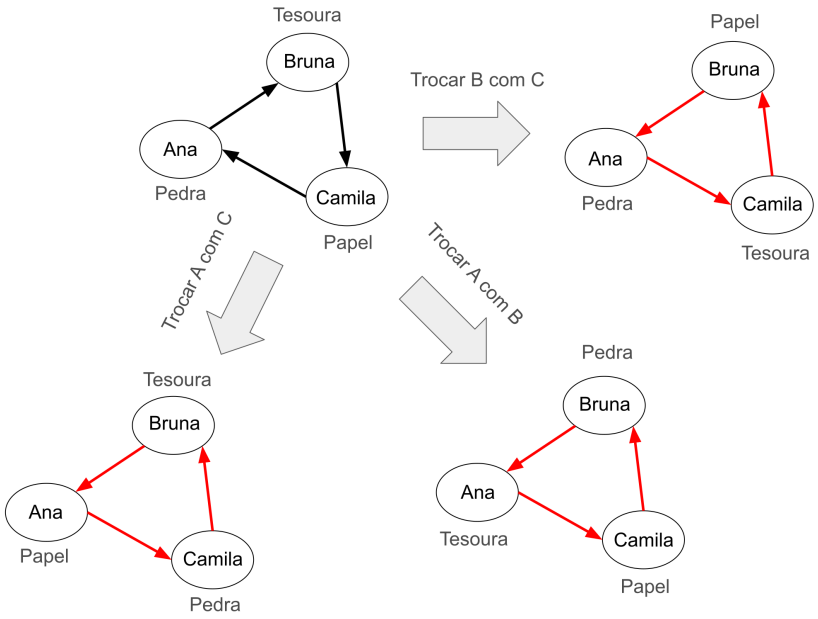
10. Pedra, Papel, Tesoura, Troca (Resolução)

Solução

(D)

Resolução

Para resolver este problema, é preciso reparar que, quando ocorre uma troca, independentemente de que carta está com cada jogador, todas as vitórias ficam invertidas.
[inserir imagem]



Nesta figura, as setas indicam como se resolve uma partida. Assim, após um número **par** de trocas, as vitórias mantêm-se iguais à situação inicial. Pelo contrário, após um número **ímpar** de trocas, as vitórias ficam invertidas.

Como a Bruna quer ganhar à Camila e isso acontece na situação inicial, ela só precisa de garantir que ocorre um número **par** de trocas.

Isto é Pensamento Computacional!

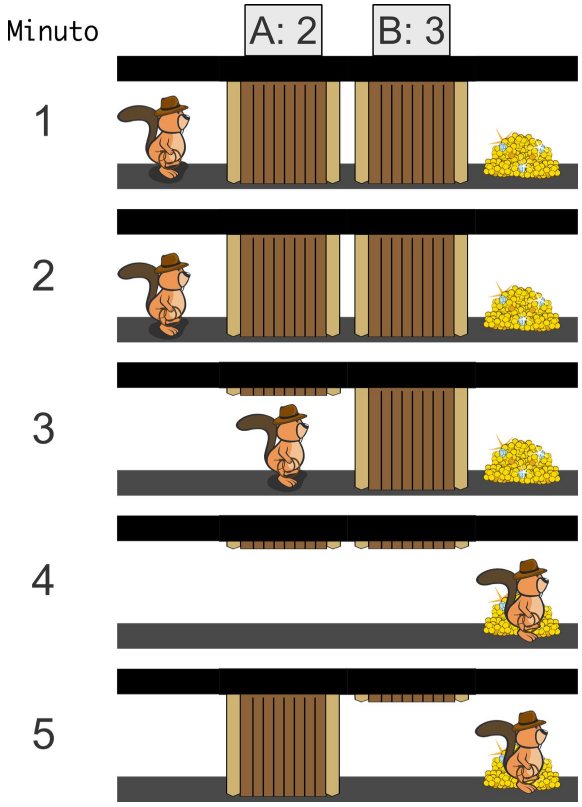
O reconhecimento de padrões é uma competência fundamental para resolver esta tarefa. À primeira vista, a tarefa pode parecer muito difícil porque parece haver demasiados resultados possíveis a considerar. No entanto, se forem analisados sistematicamente alguns casos, não é difícil perceber que afinal não há assim tantas variações e que os casos existentes podem ser agrupados de acordo com a **paridade** do número de trocas.

11. A Pirâmide dos Tesouros

O castor Jaime está numa pirâmide perigosa, que tem muitos corredores com armadilhas. No final de cada corredor, há um tesouro escondido. O castor Jaime quer chegar a qualquer tesouro o mais rapidamente possível.

Cada corredor é protegido por uma fila de blocos. Inicialmente, todos os blocos estão para baixo. Assim que alguém chega a um corredor, os seus blocos começam a mover-se periodicamente. Um bloco com período 2 sobe após 2 minutos, cai de novo após mais 2 minutos, e assim sucessivamente.

A imagem abaixo representa um exemplo de um destes corredores. Este corredor tem dois blocos, etiquetados A e B, com períodos 2 e 3, respetivamente. A imagem mostra a disposição dos blocos do corredor nos minutos 1 a 5 após o castor Jaime ter chegado.



Para chegar ao tesouro o mais rapidamente possível, o castor Jaime espera 2 minutos, dirige-se ao bloco A, espera mais 1 minuto e depois passa o bloco B para alcançar o tesouro após 3 minutos no total. Neste caso, o castor Jaime age como se seguisse esta sequência de instruções:

```
esperar(2)
ir_para_bloco(A)
esperar(1)
ir_para_tesouro
```

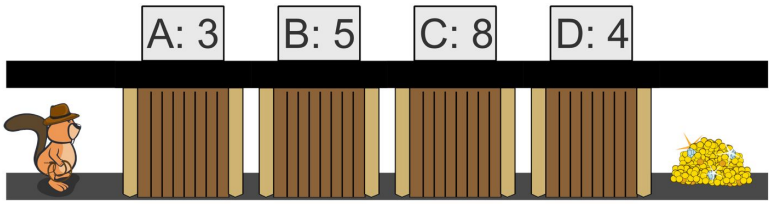
O castor Jaime poderia ter chegado ao tesouro ao mesmo tempo, mas com uma sequência de instruções mais curta:

```
esperar(3)
ir_para_tesouro
```

Assume que só podes usar os três tipos de instruções indicados na página anterior (`esperar()`, `ir_para_bloco()` e `ir_para_tesouro`).

Pergunta

O próximo corredor tem quatro blocos com períodos 3, 5, 8 e 4.



Qual é o comprimento da sequência de instruções mais curta que permite ao castor Jaime chegar ao tesouro o mais rápido possível?

Resposta

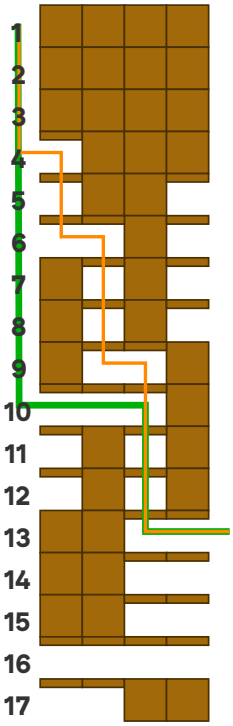
Escreve um número inteiro (entre 0 e 99) na folha de respostas.

11. A Pirâmide dos Tesouros (Resolução)

Solução

4

Resolução



Observando a imagem, é possível perceber como o castor Jaime pode mover-se por pelo corredor representado. Para se mover para uma nova posição assim que tal for possível, ele pode seguir esta sequência de instruções e chegar ao tesouro após 12 minutos (linha laranja):

```
esperar(3)
ir_para_bloco(A)
esperar(2)
ir_para_bloco(B)
esperar(3)
ir_para_bloco(C)
esperar(4)
ir_para_tesouro
```

No entanto, ele pode seguir menos instruções e chegar ao tesouro também após 12 minutos (linha verde):

```
esperar(9)
ir_para_bloco(C)
esperar(3)
ir_para_tesouro
```

Não há forma do castor Jaime seguir menos instruções sem perder tempo! Ele só poderia usar menos instruções se atravessasse o corredor de uma só vez, mas a única oportunidade para isso surge depois de esperar 15 minutos. Nesse caso, ele poderia alcançar o tesouro com apenas duas instruções – mas demoraria 15 minutos, ou seja, não seria o mais rápido possível.

Isto é Pensamento Computacional!

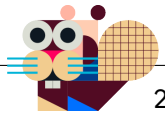
Esta tarefa abrange três aspetos importantes da informática:

Algoritmo
O conceito de algoritmo é talvez a noção mais central da ciência de computadores; foca-se na questão de como um problema pode ser resolvido e, subsequentemente, como a solução pode ser descrita de forma inequívoca. Encontrar um algoritmo requer criatividade, em contraste com a execução de um algoritmo: depois de um algoritmo estar precisamente definido, por exemplo como uma sequência de instruções numa linguagem de programação, uma máquina pode seguir essas instruções e executá-las sem qualquer compreensão.

Problema de Otimização
Esta tarefa Bebras envolve considerar diferentes abordagens para um problema, compará-las entre si e escolher a melhor abordagem de acordo com algum critério de qualidade. Em ciência de computadores, problemas deste tipo são chamados de problemas de otimização, porque o objetivo é minimizar um fator específico (neste caso, o tempo que o castor Jaime demora a chegar ao tesouro, bem como o número de instruções que segue). Um método que pode ser usado para resolver problemas de otimização é a programação dinâmica. Esta abordagem envolve decompor o problema em partes e resolver continuamente partes maiores do problema original

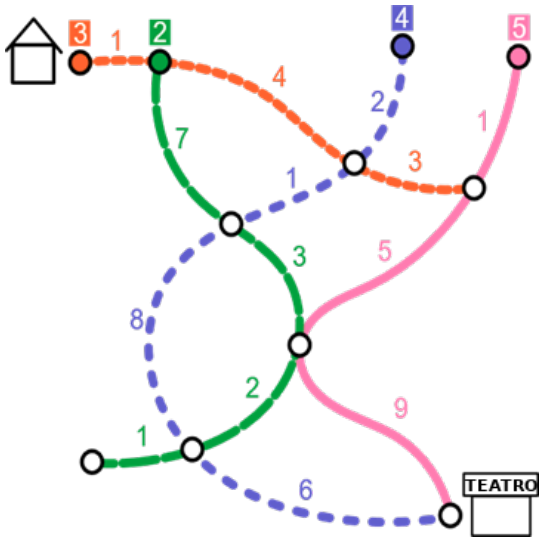
utilizando soluções parciais armazenadas. Aqui, os subproblemas parciais poderiam consistir em chegar apenas ao primeiro, segundo, terceiro ou quarto bloco antes de determinar finalmente o caminho para o tesouro.

Programação
Para tornar algoritmos executáveis por computadores, eles precisam de ser expressos numa linguagem de programação. Esta linguagem fornece um conjunto de operações fundamentais (ou comandos), como os comandos `esperar(n)` ou `ir_para_bloco(b)` neste caso. Algoritmos complexos podem ser expressos combinando comandos fundamentais com construções da linguagem que permitem execução sequencial, repetida ou condicional.



12. Transportes Públicos

O Marco quer ir de casa ao teatro de autocarro. Existem 4 linhas de autocarro unidirecionais a operar na cidade. As paragens de autocarro estão assinaladas com círculos de contorno preto. As linhas de autocarro estão representadas com cores diferentes. As paragens coloridas indicam o ponto de partida de cada linha.



O primeiro autocarro de cada linha parte da respetiva paragem inicial ao mesmo tempo. Depois, cada linha envia um autocarro em intervalos de tempo diferentes. Os números em fundos coloridos indicam o intervalo entre partidas dos autocarros, em minutos. Por exemplo, a linha rosa (linha contínua) envia um autocarro a cada 5 minutos – nos minutos 0, 5, 10, 15 e assim sucessivamente.

Os números junto aos segmentos das linhas indicam quantos minutos um autocarro demora a percorrer a distância entre duas paragens. Parar numa paragem e embarcar passageiros não demora tempo (0 minutos).

As paragens onde duas ou mais linhas se cruzam podem ser usadas para fazer uma troca de autocarro. Se o Marco chegar a um ponto de interseção, ele pode mudar para um autocarro que lá chegue mais tarde ou ao mesmo tempo que aquele em que ele se encontra.

Pergunta

Se o Marco apanhar o primeiro autocarro laranja no minuto 0, qual a menor quantidade de tempo, em minutos, que demora a chegar ao teatro?

Resposta

Escreve um número inteiro (entre 0 e 99) na folha de respostas.

12. Transportes Públicos (Resolução)

Solução

20 minutos

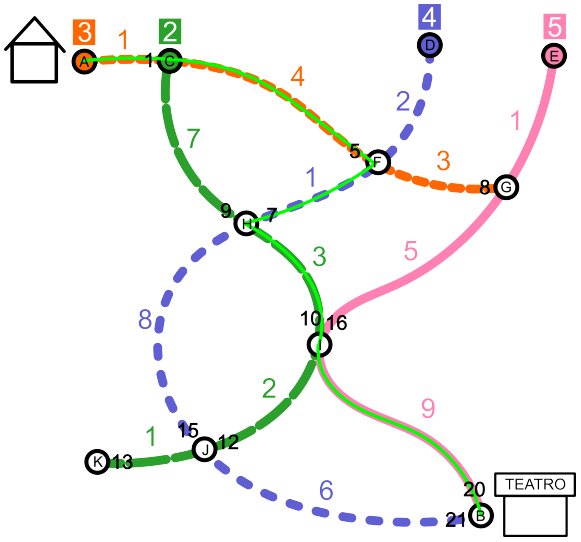
Resolução

O Marco tem de apanhar o autocarro laranja durante 2 paragens (5 minutos), depois o autocarro azul durante 1 paragem (1 minuto de espera + 1 minuto de viagem), o autocarro verde durante 1 paragem (3 minutos) e o autocarro rosa durante 1 paragem (1 minuto de espera + 9 minutos de viagem).

A resposta pode ser encontrada escolhendo uma paragem cujo tempo mais rápido para a atingir já seja conhecido (no início, apenas a paragem A). Depois, podem ser calculados os tempos das paragens alcançadas a partir dessa paragem. Este processo é repetido com diferentes paragens até que seja encontrado o tempo mais rápido para chegar ao destino.

Cada paragem pode ter múltiplas rotas possíveis de chegada com tempos diferentes, mas apenas a mais rápida é considerada. Só quando todas as possibilidades de chegada forem testadas e o tempo mais rápido encontrado é que a paragem pode ser usada para calcular os tempos das paragens alcançadas a partir dela.

A aplicação deste método no problema está descrita abaixo:



A rota correta está destacada na figura. Os números junto de cada paragem indicam o tempo mais curto necessário para chegar à paragem a partir de cada direção.

A paragem C é alcançada em 1 minuto ao apanhar o primeiro autocarro laranja. Da mesma forma, a paragem F é alcançada em 5 minutos e a paragem G em 8 minutos. A paragem I pode ser alcançada a partir da paragem G, apanhando o terceiro autocarro rosa (3 minutos de espera), totalizando 8 + 3 + 5 = 16 minutos.

A paragem H pode ser alcançada a partir da paragem C (após chegar ao minuto 1) com o segundo autocarro verde, depois de esperar 1 minuto e percorrer 7 minutos, totalizando 1 + 1 + 7 = 9 minutos desde o início. A paragem H também pode ser alcançada a partir da paragem F (após chegar ao minuto 5) com o segundo autocarro azul em 1 minuto, após 1 minuto de espera, totalizando 5 + 1 + 1 = 7 minutos. Este tempo é mais rápido que os 9 minutos calculados anteriormente, e este novo valor é usado para cálculos subsequentes. Esta chegada mais rápida permite ainda apanhar o primeiro autocarro verde para chegar à paragem I em 7 + 3 = 10 minutos. O novo tempo para a paragem I é mais rápido que os 16 minutos calculados anteriormente, pelo que os 16 minutos são descartados.

Como esta é a rota mais rápida para I, o mesmo autocarro pode ser usado para chegar à paragem J em 10 + 2 = 12 minutos, e à paragem K em 12 + 1 = 13 minutos. Alternativamente, a paragem J pode ser alcançada a partir da paragem H em 7 + 8 = 15 minutos, o que é mais lento e não é considerado.

A partir da paragem I, o segundo autocarro rosa pode ser usado para chegar à paragem B (destino) após 1 minuto de espera, totalizando 10 + 1 + 9 = 20 minutos. Este ainda não é o resultado final, porque a paragem B também pode ser alcançada a partir da paragem J. Isto seria feito com o segundo autocarro azul, após 3 minutos de espera e 6 minutos de viagem, totalizando 12 + 3 + 6 = 21 minutos. Este é mais lento que 20 minutos, pelo que 20 minutos calculados anteriormente é, de facto, a solução correta.

Isto é Pensamento Computacional!

Esta tarefa promove o pensamento computacional ao enfatizar a interpretação de dados, o raciocínio algorítmico e a sequenciação. Envolve a análise de dados estruturados, incluindo intervalos de partida dos autocarros, tempos de viagem e condições de transferência. Identificar padrões nestes dados numéricos, acompanhar múltiplas sequências sincronizadas e determinar transições válidas com base em restrições pré-definidas é essencial para encontrar a solução correta. Um processamento eficaz dos dados garante a precisão na tomada de decisões.

O pensamento algorítmico é fundamental, uma vez que a resolução da tarefa requer a construção de uma sequência estruturada de passos. Reconhecer padrões repetitivos nas partidas dos autocarros, aplicar regras condicionais para transferências válidas e avaliar sistematicamente as rotas possíveis refletem princípios algorítmicos essenciais. Esta abordagem está intimamente ligada a aplicações do mundo real, como sistemas de planeamento de horários e algoritmos de determinação de percursos.

A sequenciação é crucial, pois a ordem correta dos eventos deve ser mantida. Alinhar diferentes intervalos de tempo e durações de viagem de forma estruturada garante que a solução segue um fluxo lógico. Ao integrar a interpretação de dados com o raciocínio algorítmico, a tarefa reforça princípios de informática essenciais para automação, logística e problemas de otimização.