- Problema A - Ácido Ribonucleico Alienígena

  Leia a fita da esquerda para a direita e, onde for possível fazer uma dobra, faça-a. Usar uma pilha é ideal para a implementação: insira os caracteres sequencialmente na pilha e, quando o caractere do topo for compatível com o próximo a ser lido, você "liga" os dois, eliminando-os.

- Problema B - Dobradura

  A resposta é $(2^n + 1)^2$.

- Problema C - Building Design

  Para que o prédio seja o maior possível, sempre devemos escolher como próximo andar o maior dentre os que têm a cor apropriada e que é menor do que o andar atual. Para encontrar o maior prédio possível, basta ordenar o vetor segundo o módulo dos números dados e aplicar essa estratégia para os dois casos possíveis: quando o primeiro andar é vermelho e quando o primeiro andar é azul.

- Problema D - RSA

  Para começar, precisamos fatorar N. Como é garantido que ele é um produto de dois primos, basta fazer $\mathcal{O}(\sqrt{N})$ iterações a partir de 2 para encontrar esses dois fatores, P e Q. Calcule $\varphi(N)$ conforme descrito no enunciado e, agora, calcule D, o inverso multiplicativo modular de E módulo $\varphi(N)$, usando o Algoritmo de Euclides Estendido. A implementação desse algoritmo pode ser feita de forma bem compacta em C:

```c
/* Retorna o inverso de a modulo m */
long long inv(long long a, long long m) {
  return a<2?a:((1-m*inv(m % a,a))/a % m + m) % m;
}
```

  Agora, só falta calcular a resposta $M = C^D \mod N$, mas, atenção, fazer $\mathcal{O}(D)$ multiplicações aqui é muito ineficiente. É possível computar essa exponenciação em $\mathcal{O}(\log D)$ operações usando a seguinte relação de recorrência:

$$x^n = \begin{cases} x^{n/2} \cdot x^{n/2}, & \text{se } n \text{ é par,} \\ x \cdot x^{n-1}, & \text{caso contrário.} \end{cases}$$

- Problema E - Army Buddies

  Mantenha dois vetores E e D em que E_i é o índice do próximo soldado à esquerda de i que ainda está vivo e D_i é o índice do próximo soldado à direita de i que ainda está vivo. Assim, cada atualização na lista pode ser processada em $\mathcal{O}(1)$ e a complexidade total do algoritmo é $\mathcal{O}(S + B)$.

- Problema F - Cows and Cars

  A probabilidade de você ter escolhido uma vaca a princípio e mudado pra um carro é

$$P_1 = \frac{NCOWS}{NCOWS + NCARS} \cdot \frac{NCARS}{NCOWS + NCARS - NSHOW - 1}$$

  e a probabilidade de você ter escolhido um carro a princípio e mudado pra outro carro é

$$P_2 = \frac{NCARS}{NCOWS + NCARS} \cdot \frac{NCARS - 1}{NCOWS + NCARS - NSHOW - 1}.$$

  A resposta final é a probabilidade de ter acabado escolhendo um carro, não importando o que você escolheu no começo, ou seja, é simplesmente $P_1 + P_2$.

- Problema G - Bolsa de Valores

  Simule as negociações mantendo os valores de proposta de venda em uma *heap* e os valores de proposta de compra em outra.

- Problema H - Jumping Mario

  Faça exatamente o que pede o enunciado: conte quantos elementos da sequência são maiores e quantos são menores que o antecessor.

- Problema I - Os Joguinhos de Ramsés

  Esse problema pode ser resolvido com programação dinâmica (PD). Existem várias formas de representar o estado da PD, mas eu sugiro usar uma máscara de N bits para marcar quais blocos já foram utilizados e quais ainda estão disponíveis para serem empilhados, um inteiro representando o índice do bloco que está atualmente no topo da pilha, e outro indicando qual a dimensão – x, y ou z – que está sendo usada como altura.

  As transições de estado são simples: teste todos os blocos que ainda não foram usados e têm dimensões compatíveis com o bloco do topo da pilha, escolhendo o melhor resultado dentre todos os testados.

- Problema J - Colisão Galática

  Monte um grafo em que cada estrela é um vértice e crie uma aresta ligando todos os pares de estrelas que devem estar, obrigatoriamente, em galáxias diferentes, isto é, aqueles pares separados por uma distância menor que ou igual a 5 anos-luz. Agora, para cada componente conexa do grafo, colora seus vértices de preto ou branco, de forma que dois vértices adjacentes não tenham a mesma cor, e some min{número de vértices brancos, número de vértices pretos} à resposta.

Na hora de montar o grafo, cuidado para não estourar o limite de tempo, medir a distância entre todos os $\mathcal{O}(N^2)$ pares de vértices é muito custoso. Eu sugiro que você crie um vetor para cada coordenada x e insira nele as coordenadas em y correspondentes ao valor da abscissa que ele representa. Assim, para cada ponto $P = (X, Y)$, basta fazer algumas buscas binárias pelos vetores que guardam as coordenadas dos pontos com abscissa em $X - 5, X - 4, ..., X + 4, X + 5$ para encontrar todos os pontos que deverão ser ligados a P por uma aresta.