

## Instruções Gerais

- Esta atividade deve ser resolvida **individualmente**.
- Os itens teóricos devem resolvidos de forma organizada, clara e formal.
- A solução encontrada deve ser submetida, em um único arquivo PDF, no moodle. Certifique-se de que todas as resoluções digitalizadas estão legíveis antes de submetê-las.
- Entregas após o prazo estabelecido no moodle serão desconsideradas.
- É permitida a consulta a livros e outros materiais, mas a atividade apenas pode ser discutida com a equipe de ensino.
- Os algoritmos desenvolvidos nos itens práticos devem ser organizados e comentados. Todos os códigos utilizados devem ser submetidos como anexos no moodle.
- Qualquer tentativa de fraude, se detectada, implicará na reprovação (com nota final 0.0) de todos os envolvidos, além das penalidades disciplinares previstas no Regimento Geral da Unicamp (Arts. 226 – 237).

## Apresentação

Uma fábrica produz barras de um tamanho fixo  $n \in \mathbb{N}$  e as vende neste tamanho ou em tamanhos menores, que podem ser qualquer inteiro no conjunto  $i \in \{1, \dots, n\}$ . As barras menores são produzidas por meio do corte da barra maior e o *padrão de corte* a ser adotado pode produzir barras menores de diversos tamanhos, como ilustrado na Figura 1.

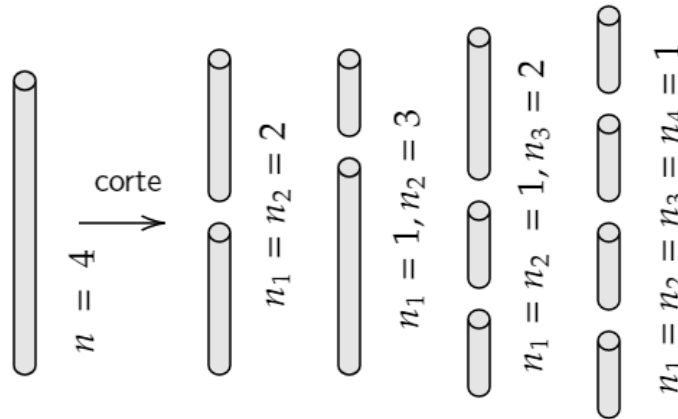


Figura 1: Possíveis padrões de corte para uma barra de tamanho  $n = 4$ .

Devido a oscilações de relações de oferta e demanda no mercado, os preços por tamanhos específicos de barras podem flutuar e, a cada mês, a fábrica planeja a sua produção a fim de maximizar o seu retorno esperado. O problema a ser resolvido nesta atividade é o seguinte: dada uma barra de tamanho  $n \in \mathbb{N}$  e uma tabela de lucros  $\ell(i)$  associados a barras de tamanho  $i \in \{1, \dots, n\}$ , determine o padrão de corte da barra que maximiza o lucro total obtido com a sua venda. Por exemplo, para a barra de tamanho  $n = 4$ , considere os dados na Tabela 1.

$i$	1	2	3	4
$\ell(i)$	4	8	13	15

Tabela 1: Dados para a barra de tamanho  $n = 4$ .

Neste caso, o padrão ótimo de corte é: um corte de tamanho 1 e um corte de tamanho 3.

## Questões

- **Questão 1:** Modele este problema como um problema de otimização, identificando:
- as variáveis de decisão;
  - a função objetivo;
  - as restrições envolvidas, incluindo as de integralidade, se houver.
- **Questão 2:** Em uma estratégia de busca por soluções de forma *enumerativa*, podemos enumerar os padrões de corte e avaliar o seu lucro obtido. Pensando que uma barra de tamanho  $n$  pode (ou não) ser cortada em  $n - 1$  posições distintas (cada uma espaçada de 1 unidade de medida), quantas soluções devem ser testadas (considere repetições)?
- **Questão 3:** Projete, por indução, um algoritmo recursivo que encontre a solução ótima para este problema. Para tanto, as seguintes perguntas devem ser respondidas pelo seu projeto:
- Qual é o caso base para a recursão?
  - Como construir a solução ótima para a barra de tamanho  $k$  se conhecermos as soluções ótimas para as barras de tamanho  $1, \dots, k - 1$ .
  - Ao final da execução, como reconstruir a solução ótima? Lembre de memorizar a melhor escolha a cada passo!
- **Questão 4:** Um problema que pode acontecer em algoritmos recursivos como o projetado acima é o re-cálculo de soluções de subproblemas durante as chamadas recursivas. Isto é, por exemplo, para resolver o problema com  $n = 4$ , o problema com  $n = 2$  é resolvido em mais de uma chamada recursiva. Isso acontece com o seu algoritmo? Se isto acontece, o seu algoritmo pode facilmente se aproximar da simples enumeração e testagem de soluções.
- **Questão 5:** Para evitar o recômputo de soluções já calculadas, proponha uma **forma ordenada** de resolver o problema, partindo de cortes menores e construindo a solução ótima para problemas maiores. Use esta forma de resolução para remover as chamadas recursivas, reescrevendo-o usando laços (*loops*). Quantas operações são necessárias, nesta versão, para resolver o problema?
- **Questão 6:** Use o algoritmo projetado para resolver o caso de teste fornecido no moodle.