



Especificação do Trabalho

1 Problema

Compressão de dados é o processo de codificar ou converter dados de uma maneira que esses ocupem menos espaço na memória (disco). A compressão pode ser feita com ou sem perdas. O presente trabalho consiste na implementação de um algoritmo sem perdas que será descrito a seguir. Este algoritmo deverá ser implementado no exercício no run.codes. Após a implementação, o programa deverá verificar se uma entrada codificada realmente é válida ou não.

2 Algoritmo

O algoritmo deverá seguir os seguintes passos:

1. Crie uma lista de probabilidades para um dado conjunto de símbolos.
2. Ordene a lista de probabilidades em ordem decrescente.
3. Divida a lista em duas partes, com a probabilidade total de ambas as partes sejam o mais próximo possível da outra.
4. Atribua o valor 0 (zero) para o lado esquerdo e o valor 1 (um) para o lado direito.
5. Repita os passos 3 e 4 até que as listas fiquem com apenas um único elemento.

3 Exemplo

A Figura 1 apresenta um exemplo do funcionamento do algoritmo:

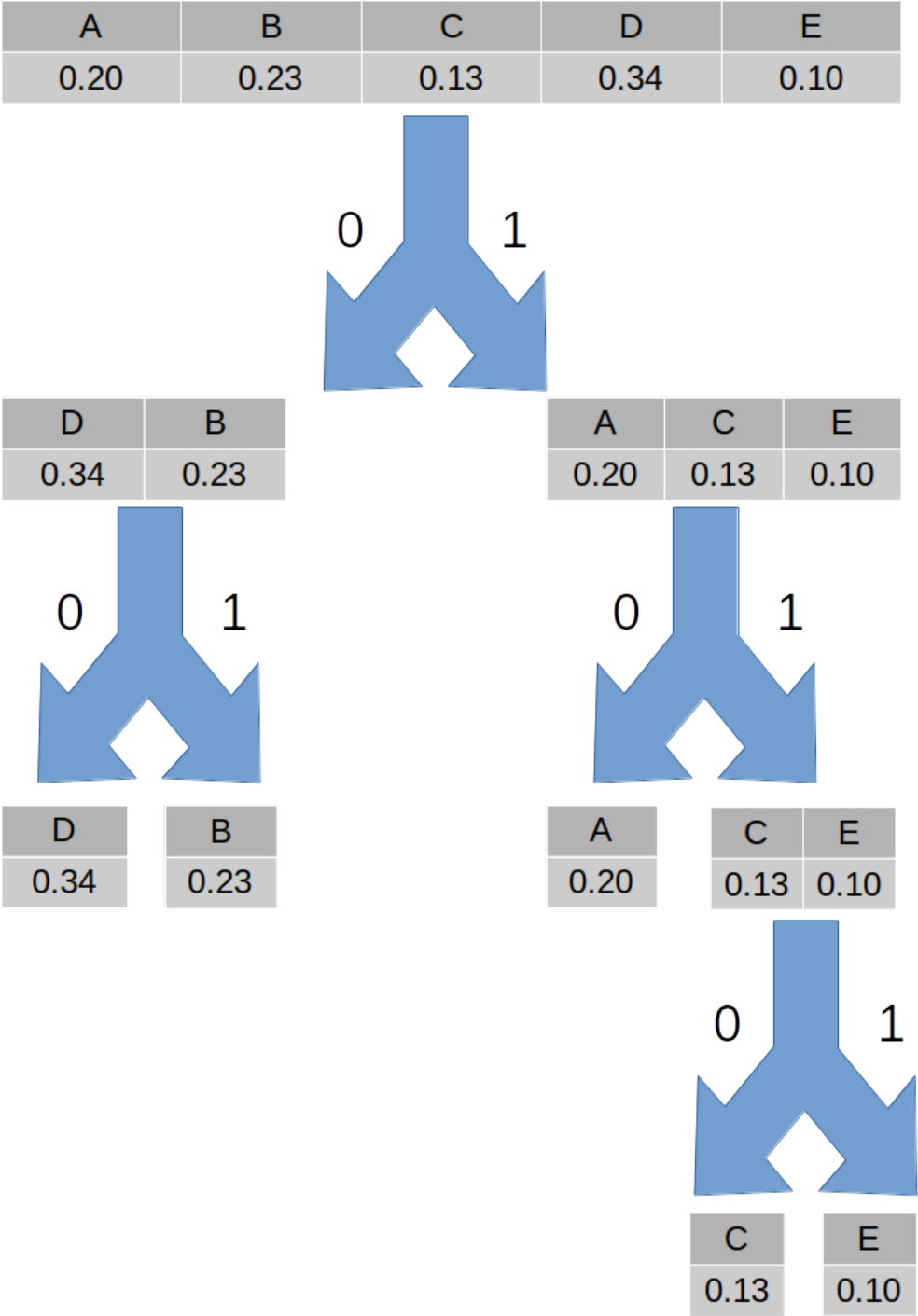


Figura 1: Codificação.

A codificação do alfabeto será:

Tabela 1:

A	B	C	D	E
0.20	0.23	0.13	0.34	0.10
10	01	110	00	111

4 Entrada de Dados

O arquivo de entrada poderá ter diversas linhas na seguinte ordem. A primeira linha representa a quantidade de N elementos. A seguir virá uma lista com N elementos e as suas probabilidades. Após isso, a última linha terá um código binário que deverá ser validado ou não conforme o código de cada letra.

5 Saída de Dados

A saída deverá ser **sim**, quando o termo for válido, e **nao**, quando termo não for válido.

6 Exemplo

6.1 Exemplo - Entrada

```
5
A 0.20
B 0.23
C 0.13
D 0.34
E 0.10
1001
```

6.2 Exemplo - Saída

```
sim
```

7 Prazos

- Início: 22/12/2022 às 10:00 horas (horário do servidor).
- Encerramento: 03/01/2023 às 10:00 horas (horário do servidor).

8 Regras gerais

- Lembrem-se, a entrada de dados é feita via `input` e a saída via `print`.
 - Não é necessário mostrar mensagem para entrada de dados. A saída deve apresentar apenas a resposta sem mensagem adicional.
 - Na construção do seu programa você deve usar apenas os conceitos aprendidos em sala de aula. Respostas que utilizem bibliotecas prontas não serão consideradas.
 - Em caso de plágio, **todos** os envolvidos receberão **nota zero!**
 - Na execução do seu programa no `run.codes`, existem casos de testes que vão além dos exemplos mostrados a seguir. Esses casos de teste não serão revelados. Pense em exemplos de entradas e saídas que podem acontecer e melhore o seu código para capturá-las.
 - Cada aluno deve ter **apenas uma conta**. Se houver duplicidade de resposta do mesmo aluno, será atribuída **nota zero!**
 - Esse é um trabalho prático e o resultado fará parte no cômputo da nota da disciplina.
 - A interpretação do problema faz parte da avaliação. Você deve fazer o problema sozinho. Evite discutir a solução com outras pessoas, **especialmente com outros colegas de sala de aula**.
 - A legibilidade do código-fonte será também considerada na avaliação.
 - Se for necessário, o professor poderá arguir a defesa do trabalho submetido ao aluno. Essa arguição deverá ocorrer de forma presencial.
 - Os alunos que não tiverem acesso ao computador em casa deverão utilizar os laboratórios de informática disponibilizados na Escola Superior de Tecnologia (EST).
-