

## Universidade Federal do Rio Grande do Norte

**Alunos:** Gustavo Henrique Miguel Ferreira, Rodrigo Nunes de Castro, Danilo Damasceno

**Matérias:** Estrutura de Dados Básicas II e LPII

**Atividade:** Projeto da Terceira Unidade

# Manual de Programador

O projeto consiste em verificar qual a melhor combinação de N casas com d ligações e uma matriz com os custos relacionados as ligações de uma casa a outra. O projeto foi desenvolvido com as seguintes estruturas:

- **Classes**

Nosso projeto possui seis classes: Casa, Celula, Combinacao, Disjoint, Wiriter e TrabalhoDeEDBII. Vamos dar uma breve comentada sobre o papel de cada uma.

- Casa:

É nessa classe que acontece todo o processo de receber as combinações e verificar se são válidas, calcular o melhor custo, calcular o total de combinações válidas, guardar a combinação de melhor custo. E nessa mesma classe copio os valores da matriz de custos do arquivo txt para um campo matriz de custos na classe.

- Combinacao:

Nessa classe foi utilizado o código encontrado no seguinte site: <http://stackoverflow.com/questions/127704/algorithm-to-return-all-combinations-of-k-elements-from-n> e adaptado para a necessidade do projeto. Essa classe gera todas as combinações de acordo com a quantidade de casas e as possíveis ligações.

- Disjoint:

Classe que implementa o conjunto disjunto com os métodos, union, unionByRank, Find, makeSet.

- Celula:

Classe com dois atributos do tipo inteiro, um atributo armazena a linha da matriz e o outro a coluna. Serve para fazer referência a matriz de custos.

- Writer:

Classe com um único método que tem a função de escrever no arquivo de saída.

- TrabalhoDeEDBII:

Classe que contém o main.

- **Estruturas de dados Utilizadas:**

Nesse projeto usamos vetores e conjuntos disjuntos, os vetores foram usados para armazenar os valores dos custos, o vetor de grau( o grau de conexão de cada casa), matriz de custos e na parte de combinação também foi bastante utilizado. Os conjuntos disjuntos foram utilizados para Verificar se a ligação formava algum ciclo e fazer a união dos elementos que podem ser ligados respeitando as condições.

- **Problemas enfrentados no decorrer do projeto:**

Um problema que tivemos foi com a parte de combinação, inicialmente foi feito um algoritmo para gerar as combinações e que ao final não corresponde ao que queremos e por questão de tempo resolvemos adaptar uma solução que encontramos na internet.

A nossa classe de combinação tem um problema quando se trata de gerar as combinações a partir de nove casas, como o valor é bastante alto não conseguimos gerar o valor(o número de combinações) para criar as combinações.

Outro problema é em relação ao total de combinações válidas, ele não gera o número correto, pois no algoritmo acaba contando alguns casos válidos a mais.

- **Análise assintótica**

Não conseguimos calcular exatamente a complexidade do algoritmo, mas pelos testes que realizamos vemos que o mesmo é na ordem de complexidade exponencial.

- **Diagrama de Classes**

