

Trabalho Prático 1

Álvaro Cândido de Oliveira Neto

Universidade Federal de Minas Gerais (UFMG)

Belo Horizonte – MG – Brasil

djalv009@ufmg.br

1. Introdução

Devido a alguns problemas logísticos no aluguel das bicicletas, foi nos pedido para desenvolver o aplicativo +Bikes, que realiza a alocação das bicicletas de acordo com a distância e preferência dos visitantes interessados.

2. Modelagem computacional

Para resolver o problema foi quebrado em três partes:

- 1- Calcular a distância das bikes até os visitantes;
- 2- Ordenar a lista de preferências dos visitantes;
- 3- Realizar a alocação das bikes.

Para calcular a distância das bikes, foi criado um grafo não direcionado e foi usado o algoritmo de busca em largura (BFS) que percorre o grafo a partir de um nó fonte e armazena as camadas de cada nó vizinho do nó fonte, e cada camada corresponde à distância.

Para ordenar a lista de preferências dos visitantes foi usado o Insertion sort.

E por fim, para realizar a alocação das bikes foi usado o algoritmo de Gale-Shapley, que realiza a alocação estável para todas as bikes e vizinhos seguindo as listas de preferências calculadas anteriormente.

3. Estruturas de dados e algoritmos

Para armazenar a lista de adjacências do grafo foi utilizado as listas e para armazenar a lista de preferências dos visitantes e a distância das bikes foi utilizado vetores, todas as estruturas de dados citados acima já estavam implementadas na STL do C++, os algoritmos utilizados foram o de busca em largura (BFS) e o Gale-Shapley.

Pseudocódigos:

```
BFS(Node s) {
    Defina todos os nós como Brancos;
    Seja q uma fila vazia;
    Seja d um array vazio;

    s.color = "Cinza";
    s.dist = 0;
    d[s] = s.dist;

    q.enqueue(s);

    While(q é não vazia) {
        u = q.dequeue();

        Para cada vizinho v de u {
            Se v for Branco {
                v.color = "Cinza";
                v.dist = u.dist + 1;
                d[v] = v.dist;
                q.enqueue(v);
            }
        }
        u.color = "Black";
    }
    return d;
}

Gale-Shapley() {
    Seja s uma pilha com os visitantes sem bikes;

    While(q é não vazia) {
        v = q.pop();

        seja b a próxima bike a ser proposta por v em sua lista preferencial

        if(b não tiver par) {
            (v,b) formarão um novo par;
        }
        else {
            Seja v' o atual par de b;
            if(b prefere v a v')
                (v,b) formarão par;
                q.push(v');
            else {
                q.push(v);
            }
        }
    }
}
```

```
}  
    }  
}
```

4. Análise Assintótica

O algoritmo de Gale-Shapley, tirando a função de impressão, roda em tempo **$O(n^2)$** , onde n é a quantidade de bikes e visitantes, pois é todas as combinações possíveis.

O algoritmo de busca em largura roda em tempo **$O(v+a)$** , onde v é quantidade de vértices e a é a quantidade de arestas, só é possível esse tempo pois o grafo foi implementado usando uma lista de adjacências.

Para criar as listas de preferências leva um tempo de **$O(n^3)$** , pois temos que ordenar uma matriz $n \times n$.

5. Instruções de compilação e execução

1. No terminal digite: make
2. Ainda no terminal digite: ./bin/tp01 nome_do_arquivo.extensão