

Aluno: Jefferson Serafim Ascano NUSP 6431284  
Aluno: Wallace Faveron de Almeida NUSP 6879609  
MAC0438 - 2011 - IME-USP  
Prof. Daniel Macedo Bastista

## **Relatório - EP2 – Caminhos mínimos**

A cada iteração, as threads processam caminhos de um mesmo tamanho. Processam caminhos de tamanho 1 na primeira iteração, tamanho 2 na segunda, e assim por diante. Inicialmente, existe um único "caminho" que é o que contém apenas o vértice 0.

Ao processar um caminho, a thread (função `find_path` no `main.cc`) tenta ir adicionando vértices vizinhos ao último vértice do caminho. Assim ela vai criando vários outros caminhos de tamanho maior (um vértice a mais), e verifica se o vértice de destino já possui o mínimo de caminhos que precisa. Além disso, adiciona os caminhos encontrados na lista de caminhos a serem processados (só serão processados na próxima iteração).

A barreira utilizada no exercício é do tipo barreira de disseminação. Sua implementação foi feita com base no modelo proposto na principal bibliografia do curso e nos slides das aulas. Pode ser visualizada no arquivo "`barrier.cc`", especificamente entre as linhas 5 e 16 (inicialização), e entre as linhas 22 e 28 (a barreira). Ela serve para garantir que todas as threads estão processando caminhos de mesmo tamanho, assim eles são adicionados na ordem de tamanho nos vértices de destino.

- Saída para execução do `nsfnet.txt`

*Iteracao 1: Thread 0 chegou na barreira pela 1a vez*  
*Iteracao 1: Thread 1 chegou na barreira pela 1a vez*  
*Iteracao 1: Thread 1 chegou na barreira pela 2a vez*  
*Iteracao 1: Thread 0 chegou na barreira pela 2a vez*  
*Caminhos encontrados na iteracao 1:*

*Para o vertice 1:*  
*(1) 0 - 1*

*Para o vertice 2:*  
*(1) 0 - 2*

*Para o vertice 3:*

*Para o vertice 4:*  
*(1) 0 - 4*

*Para o vertice 5:*

*Para o vertice 6:*

*Para o vertice 7:*

*Para o vertice 8:*

*Para o vertice 9:*

*Para o vertice 10:*

*Para o vertice 11:*

*Para o vertice 12:*

*Para o vertice 13:*

*Para o vertice 14:*

*Para o vertice 15:*

*Iteracao 2: Thread 0 chegou na barreira pela 3a vez*

*Iteracao 2: Thread 1 chegou na barreira pela 3a vez*

*Iteracao 2: Thread 1 chegou na barreira pela 4a vez*

*Iteracao 2: Thread 0 chegou na barreira pela 4a vez*

*Caminhos encontrados na iteracao 2:*

*Para o vertice 1:*

*(1) 0 - 1*

*(2) 0 - 2 - 1*

*Para o vertice 2:*

*(1) 0 - 2*

*(2) 0 - 1 - 2*

*Para o vertice 3:*

*(2) 0 - 1 - 3*

*(2) 0 - 4 - 3*

*Para o vertice 4:*

*(1) 0 - 4*

*Para o vertice 5:*

*(2) 0 - 4 - 5*

*Para o vertice 6:*

*(2) 0 - 2 - 6*

*Para o vertice 7:*

*Para o vertice 8:*

*(2) 0 - 1 - 8*

*Para o vertice 9:*

*Para o vertice 10:*

*Para o vertice 11:*

*(2) 0 - 4 - 11*

*Para o vertice 12:*

*Para o vertice 13:*

*Para o vertice 14:*

*Para o vertice 15:*

*Iteracao 3: Thread 0 chegou na barreira pela 5a vez*

*Iteracao 3: Thread 1 chegou na barreira pela 5a vez*

*Iteracao 3: Thread 1 chegou na barreira pela 6a vez*

*Iteracao 3: Thread 0 chegou na barreira pela 6a vez*

*Caminhos encontrados na iteracao 3:*

*Para o vertice 1:*

*(1) 0 - 1*

*(2) 0 - 2 - 1*

*Para o vertice 2:*

*(1) 0 - 2*

*(2) 0 - 1 - 2*

*Para o vertice 3:*

*(2) 0 - 1 - 3*

*(2) 0 - 4 - 3*

*Para o vertice 4:*

*(1) 0 - 4*

*(3) 0 - 1 - 3 - 4*

*Para o vertice 5:*

*(2) 0 - 4 - 5*

*(3) 0 - 2 - 6 - 5*

*Para o vertice 6:*

*(2) 0 - 2 - 6*

*(3) 0 - 1 - 2 - 6*

*Para o vertice 7:*

*(3) 0 - 1 - 8 - 7*

*(3) 0 - 4 - 5 - 7*

*Para o vertice 8:*

*(2) 0 - 1 - 8*

*(3) 0 - 2 - 1 - 8*

*Para o vertice 9:*

*(3) 0 - 2 - 6 - 9*

*Para o vertice 10:*

*(3) 0 - 1 - 8 - 10*

*Para o vertice 11:*

*(2) 0 - 4 - 11*

*Para o vertice 12:*

*(3) 0 - 4 - 11 - 12*

*Para o vertice 13:*

*(3) 0 - 2 - 6 - 13*

*Para o vertice 14:*

*(3) 0 - 4 - 11 - 14*

*Para o vertice 15:*

*Iteracao 4: Thread 0 chegou na barreira pela 7a vez*

*Iteracao 4: Thread 1 chegou na barreira pela 7a vez*

*Iteracao 4: Thread 1 chegou na barreira pela 8a vez*

*Iteracao 4: Thread 0 chegou na barreira pela 8a vez*

*Caminhos encontrados na iteracao 4:*

*Para o vertice 1:*

*(1) 0 - 1*

*(2) 0 - 2 - 1*

*Para o vertice 2:*

*(1) 0 - 2*

*(2) 0 - 1 - 2*

*Para o vertice 3:*

*(2) 0 - 1 - 3*

*(2) 0 - 4 - 3*

*Para o vertice 4:*

*(1) 0 - 4*

*(3) 0 - 1 - 3 - 4*

*Para o vertice 5:*

*(2) 0 - 4 - 5*

*(3) 0 - 2 - 6 - 5*

*Para o vertice 6:*

*(2) 0 - 2 - 6*

*(3) 0 - 1 - 2 - 6*

*Para o vertice 7:*

*(3) 0 - 1 - 8 - 7*

*(3) 0 - 4 - 5 - 7*

*Para o vertice 8:*

*(2) 0 - 1 - 8*

*(3) 0 - 2 - 1 - 8*

*Para o vertice 9:*

*(3) 0 - 2 - 6 - 9*

*(4) 0 - 1 - 2 - 6 - 9*

*Para o vertice 10:*

*(3) 0 - 1 - 8 - 10*

*(4) 0 - 2 - 1 - 8 - 10*

*Para o vertice 11:*

*(2) 0 - 4 - 11*

*(4) 0 - 1 - 3 - 4 - 11*

*Para o vertice 12:*

*(3) 0 - 4 - 11 - 12*

*(4) 0 - 1 - 8 - 10 - 12*

*Para o vertice 13:*

*(3) 0 - 2 - 6 - 13*

*(4) 0 - 1 - 2 - 6 - 13*

*Para o vertice 14:*

*(3) 0 - 4 - 11 - 14*

*(4) 0 - 1 - 8 - 10 - 14*

*Para o vertice 15:*

*(4) 0 - 4 - 11 - 12 - 15*

*(4) 0 - 4 - 11 - 14 - 15*

*Numero de iteracoes: 4*

*As threads se encontraram na barreira 8 vezes (2 por iteracao).*

*Para o vertice 1:*

*(1) 0 - 1*

*(2) 0 - 2 - 1*

*Para o vertice 2:*

*(1) 0 - 2*

*(2) 0 - 1 - 2*

*Para o vertice 3:*

*(2) 0 - 1 - 3*

*(2) 0 - 4 - 3*

*Para o vertice 4:*

*(1) 0 - 4*

*(3) 0 - 1 - 3 - 4*

*Para o vertice 5:*

*(2) 0 - 4 - 5*

*(3) 0 - 2 - 6 - 5*

*Para o vertice 6:*

(2) 0 - 2 - 6

(3) 0 - 1 - 2 - 6

*Para o vertice 7:*

(3) 0 - 1 - 8 - 7

(3) 0 - 4 - 5 - 7

*Para o vertice 8:*

(2) 0 - 1 - 8

(3) 0 - 2 - 1 - 8

*Para o vertice 9:*

(3) 0 - 2 - 6 - 9

(4) 0 - 1 - 2 - 6 - 9

*Para o vertice 10:*

(3) 0 - 1 - 8 - 10

(4) 0 - 2 - 1 - 8 - 10

*Para o vertice 11:*

(2) 0 - 4 - 11

(4) 0 - 1 - 3 - 4 - 11

*Para o vertice 12:*

(3) 0 - 4 - 11 - 12

(4) 0 - 1 - 8 - 10 - 12

*Para o vertice 13:*

(3) 0 - 2 - 6 - 13

(4) 0 - 1 - 2 - 6 - 13

*Para o vertice 14:*

(3) 0 - 4 - 11 - 14

(4) 0 - 1 - 8 - 10 - 14

*Para o vertice 15:*

(4) 0 - 4 - 11 - 12 - 15

(4) 0 - 4 - 11 - 14 - 15