

Pesquisa em largura

		1ª iteração		2ª iteração	
a)		0 - fonte		0 - fonte	
caminho	Incremento	(1, 0/8)	(5, 0/5)	(1, 0/8)	(5, 5/5)
0,5,3,4	5	(2, 0/8)	(3, 0/10)	(2, 0/8)	Tá cheio, não passa mais fluxo
0,5,6,7,4	3	x	(4, 0/5) - dreno!	não vale a pena ver os filhos	(3, 0/8)
Final- arcos e arcos inversos		fluxo mínimo é 5		(4, 5/5) - cheio	(5, -5/0)
		(1, 3/8)	(2, 3/8)	(6, 0/7)	(7, 0/3)
		(1, -3/0)	(2, -3/8)	(4, 0/4) - dreno!	fluxo mínimo é 3
0 - fonte		(6, 3/7)	(6, -3/0)		
		(5, 5/5)			
		(5, -5/0)			

b) O fluxo máximo é $5+3=8$

arco (v1,v2)	fluxo
(0,1)	3
(0,5)	5
(1,2)	3
(5,3)	2
(5,6)	3
(2,3)	3
(3,4)	5
(6,7)	3
(7,4)	3

2) Para que $\Phi(Q)$ seja válida é preciso verificar as seguintes condições:

P1) $\Phi(Q_0)=0$ Como a fila Q, no seu estado inicial Q0 está vazia, $size=0$, então esta propriedade verifica-se.

P2) $\Phi(Q_i) \geq 0$ Após qualquer operação i, a fila Q tem zero ou mais elementos. Como é possível remover vários elementos, inclusive todos, através de uma só operação (multiDequeue), a fila pode ficar vazia. E portanto esta propriedade também se verifica.

Assim conclui-se que a função é válida.

Operação	Custo real ci	$\Delta \Phi(Q) = \Phi(Q_i) - \Phi(Q_{i-1})$	Custo amortizado $c_i + \Delta \Phi(Q)$	
size	1	$s'-s=0$	$1+0=1 O(1)$	
enqueue	1	$s'-s=(s+1)-s=1$	$1+1=2 O(1)$	
dequeue	1	$s'-s=(s-1)-s=-1$	$1-1=0 O(1)$	
multidequeue	k	$s'-s=(s-k)-s=-k$	$k-k=0 O(1)$	No pior caso removem-se todos os elementos a fila fica vazia ($s'=0$). k elementos a serem removidos. k é o mínimo entre numElems e s.

s é o nº de elementos na fila Q s' é o nº de elementos na fila Q após a operação

3) M é o nº de elementos dados no input. E portanto será a quantidade de elementos a inserir na fila e o nº existente até à remoção.

Operação	Complexidade no pior caso	
Criar a fila de Fibonacci	$\Theta(1)$	
Inicializar a fila com M elementos	$\Theta(M)$	
Criar um objecto da classe Random()	$\Theta(1)$	
Inicializar o maxDecrement	$\Theta(1)$	
Calcular M decrementos e decrementar a chave dos M elementos	$O(M)$	No pior caso ao fazer um decreaseKey todos os M elementos ficam como raízes.
Retornar o mínimo da fila	$O(M)$	No pior caso todos os M elementos são raízes e o consolidate tem de iterar sobre os M-1 elementos e criar uma única árvore.
Custo Total	$O(M)$	