sqtpm
[202619]

voltar

Trabalho: 08-heap-aumentado
Linguagens: C
Data de abertura: 2017/10/02 14:00:00
Data limite para envio: 2017/10/09 12:00:00 (encerrado)
Número máximo de envios: 25
Casos-de-teste abertos: casos-de-teste.tgz

Heap aumentado

Em várias aplicações de um heap de mínimo as operações essenciais são inserir uma chave e sua prioridade, remover a chave com prioridade mínima e reduzir o valor da prioridade de uma chave. Um exemplo importante é o algoritmo para calcular caminhos de custo mínimo em grafos, chamado de Algoritmo de Dijkstra.

Como discutimos em sala, reduzir o valor da prioridadede de uma chave requer encontrá-la. Em um heap isso só pode ser feito por uma busca seqüencial. Fazendo a busca seqüencial, reduzir o valor de uma prioridade no heap vai consumir um número de operações proporcional a n, o número de chaves no heap.

Mas, ainda como discutimos em sala, podemos aumentar o heap com um vetor, com uma tabela de hashing, com uma árvore de busca etc., de forma que a operação que diminui o valor de uma prioridade seja mais eficiente.

Seja H um heap de mínimo implementado de forma seqüencial. Vamos chamar de rank(k) a posição da chave k no vetor que guarda os nós de H. Suponha que H será aumentado por um vetor V. V é tal que V[i] = rank[i] para toda chave i em H. Todas as operações no heap devem atualizar V adequadamente. Sempre que o valor da prioridade de uma chave k precisar ser reduzido, V[k] deve ser consultado para recuperar rank(k) e acessar o nó que contém k e a prioridade de k no heap.

Dessa forma, diminuir o valor de uma chave no heap vai consumir um número de operações proporcional a log2(n) no pior caso.

Escreva um programa em C que constrói um heap de mínimo aumentado com capacidade para n pares {chaves,custo}, inicialmente vazio. As chaves são números no intervalo [0,n) e os custos são inteiros. O heap é minimo no custo.

A primeira linha da entrada contém o número n. Depois de criar o heap o programa deve processar comandos para atualizar o heap aumentado. Os comandos são:

- i k c

  Inserir o par {k,c} no heap aumentado. Se k já existir, essa operação não deve fazer nada.

- m

  Remover a chave com custo mínimo do heap e imprimir em uma linha. Se o heap já estiver vazio então o programa deve imprimir 'vazio'. Veja o exemplo abaixo.

- d k c'

sqtpm
[202619]

voltar

Diminuir o valor do custo da chave k para c'.

- t

Terminar o programa.

Exemplo

Entrada

```
25
i 10 15
i 12 6
i 13 9
i 14 18
i 15 23
i 0 50
i 15 30
d 14 -7
d 10 8
m
m
m
m
m
m
m
t
```

Saída

```
minimo {14,-7}
minimo {12,6}
minimo {10,8}
minimo {13,9}
minimo {15,23}
minimo {0,50}
vazio
```

Sugestão

- Comece implementando um heap. Depois que ele funcionar, aumente com o vetor.
