Time Complexity

initialization $O(1)$ or $O(n)$

initialize the PQ to have space for all n vertices

Each vertex

placed in Q at most once
+ removed at most once

$$O(n(\underbrace{T_I(n)}_{\text{time to insert in Q}} + \underbrace{T_E(n)}_{\text{time for extractMax in Q}}))$$

For each edge

all constant time except
update (increasePriority)

$$O(m\underbrace{T_U(n)}_{\text{time to update}})$$

Dijkstra's + Prim's Algorithm

Note Title 11/29/2007

Greedy Tree Builder

Time Complexity

initialization $O(1)$ or $O(n)$ initialize the PQ to have space for all n vertices

Each vertex

placed in Q at most once + removed at most once $\Big\}\ O(n(\underbrace{T_I(n)}_{\text{time to insert in Q}} + \underbrace{T_E(n)}_{\text{time for extractMax in Q}}))$

For each edge

all constant time except update (increasePriority) $\Big\}\ O(m\,\underbrace{T_U(n)}_{\text{time to update}})$

max # vertices in PQ

$$O(n\,\underbrace{T_I(n)}_{\text{insert}} + n\,\underbrace{T_E(n)}_{\text{extract Max}} + \underline{m}\,\underbrace{T_u(n)}_{\text{update (increase Priority)}})$$ } Adj List rep.

Binary heap

$T_I(n), T_E(n), T_u(n)$ $\quad O(\log n)$

Time $O(m \log n)$ assume $m \geq n$

Fibonacci heap

$T_I(n) + T_u(n)$ when increase priority $\quad$ amortized $O(1)$

WORST-case cost of $O(m + n \log n)$