CS 261 - Data Structure **Spring 2017**

HomeWork 5

May 12, 2017 *Liangjian Chen*

Problem 1 **Solution:**

there would be one tree with 64 nodes.

Problem 2 **Solution:**

Let's say the potential function is $f(n) = cn \log n$.

Insert operation: $\log n + f(n+1) - f(n) = \log n + c(n+1)\log(n+1) - cn\log n = O(\log n)$

Delete operation: $\log n + f(n-1) - f(n) = \log n + c(n-1)\log(n-1) - cn\log n = O(1)$

Problem 3 **Solution:**

For all even number n, n-element heap has a element with one child.

The child is lay on right most position of the last layer. This element would be on the top of his child. If consider this binary heap as an 1-index array, child would be in the n^{th} position while it's father lie on the $\lfloor \frac{n}{2} \rfloor^{th}$ position.

Problem 4 **Solution:**

Assume n,m is the number of nodes and edges in a graph. When applying Dijkstra with k-ary heap. The time complexity is $O(m\log_k n + nk\log_k n)$

Now, $n = 2^d, m = d2^{d-1}$, let's say $f(k) = d2^{d-1}\log_k 2^d + 2^d k \log_k 2^d$

$$f(k) = d2^{d-1}\log_k 2^d + 2^d k \log_k 2^d$$
$$= d2^{d-1}\frac{\log_2 2^d}{\log_2 k} + 2^d k \frac{\log_2 2^d}{\log_2 k}$$
$$= \frac{d^2 2^{d-1} + kd2^d}{\log_2 k}$$

Take derivative of $f(k)$

$$f'(k) = \frac{d2^d \log_2 k - \frac{(d^2 2^{d-1} + kd2^d)}{k \ln 2}}{\log_2^2 k}$$

assume $f'(k) = 0$, obtain that

$$\frac{d}{2} = k\ln k - k$$

the solution of k would give us the best bound

But what is the best bound? Apparently if we choose $k = d/2$, we can obtain a bound of $O(\frac{d^2 2^d}{\log d})$. Now I am going to approve this bound is same as the bound of best k. To achieve that, we need prove the following:

for a constant C and the best k_0 which follows $\frac{d}{2} = k_0 \ln k_0 - k_0$, the following inequality should holds:

$$\frac{d^2 2^{d-1} + k_0 d 2^d}{\log_2 k_0} \geq C \frac{d^2 2^d}{\log_2 d}$$

$$C \leq \min_d \left(\frac{\log d}{\log k_0} * \frac{d + 2k_0}{2d}\right)$$

$$\frac{\log d}{\log k_0} * \frac{d + 2k_0}{2d} \geq \frac{\log d}{\log k_0} * \frac{1}{2} \geq \frac{1}{2}$$

Thus there is a constant $C < \frac{1}{2}$ to achieve this inequality. Now we can say the best bound is indeed $O(\frac{d^2 2^d}{\log d})$.