15-750: Graduate Algorithms	January 29, 2017

Lecture 6: Dynamic Optimality Conjecture

Lecturer: Gary Miller *Scribe: Ilai Deutel, Maria Khutoretsky*

1 Dynamic Optimality Conjecture

Suppose we have keys $k_1, k_2, ..., k_n$ that we store in a BST and we get a sequence of requests. We want to find an algorithm that minimizes the total search time.

Definition 1.1. A Dynamic Binary Search Tree Algorithm:

1. Is based on a Binary Search Tree.
2. This Binary Search Tree is updated with rotations between searches.
3. We may look into future requests to determine rotations.

For instance, if I know that a certain user is going to use my algorithm soon, I can reorganize the whole tree in order to make it more efficient for this particular user.

Conjecture 1.2. *A Splay Tree [Sleator and Tarjan, 1985] does at most a constant factor more work than any Dynamic Binary Search Tree.*

Remark 1.3. This conjecture is still open.

In this lecture, we are going to consider special cases of this conjecture.

2 Review and Balance Theorem

Definition 2.1 (Potential function and unit cost)**.** Let T be a splay tree.

1. The weight $w(x)$ of a node $x \in T$ is 1.
2. For each node x of T, $S(x) = \sum_{y \in subtree(x)} w(y)$
3. For each node x of T, $rank(x) = \lfloor \log_2(S(x)) \rfloor$
4. The potential of the tree is $\Phi(T) = \sum_{x \in T} rank(x)$
5. $unitCost = \#$ rules applied

Our goal in this lecture is to modify this set of conditions in order to prove useful theorems.

Theorem 2.2 (Balance Theorem)**.** *The number of rotations when we perform m splays on a n-node tree is $\mathcal{O}(m \log n + n \log n)$.*

Proof. 1. According to the Corollary 4.8. to the Access Lemma (lecture 5), the amortized cost to do a splay has an upper bound: $AC \leq 3 \log n + 1$

2. $\Phi_{end} \geq 0$

3.

$$\begin{aligned}
\Phi_{begin} &= \sum_{x \in T_{begin}} rank(x) \\
&= \sum_{x \in T_{begin}} \lfloor \log_2(S(x)) \rfloor \\
&= \sum_{x \in T_{begin}} \lfloor \log_2(|subtree(x)|) \rfloor \\
&\leq \sum_{x \in T_{begin}} \lfloor \log_2(n) \rfloor \\
&\leq n \lfloor \log_2(n) \rfloor \\
&\leq n \log_2(n)
\end{aligned}$$

Therefore,

$$\begin{aligned}
\#rotations &\leq \mathcal{O}(m(3\log n + 1) + \Phi_{begin} - \Phi_{end}) \\
&= \mathcal{O}(m \log n + n \log(n))
\end{aligned}$$

□

3 Known Special Cases of Dynamic Optimality

Assumptions until the end of the lecture

1. The Binary Search Tree contains n keys $\{k_1, \cdots, k_n\}$.
2. We only perform searches on the Binary Search Tree.
3. We only search for keys that are in the Binary Search Tree (no miss).

Theorem 3.1 (Static Optimality Theorem: Version 1)**.** *If we perform m searches among the keys such that $\forall i \in \{1, ..., n\}$, k_i is searched q_i times, then:*

$$\textit{Total Cost of Splay} = \mathcal{O}\left(m + \sum_{\substack{i \in \{1,\ldots,n\} \\ q_i > 0}} q_i \log\left(\frac{m}{q_i}\right) \right)$$

Note: $m = \sum_{i=1}^{n} q_i$

Remark 3.2. $\sum_{\substack{i \in \{1,\ldots,n\} \\ q_i > 0}} q_i \log\left(\frac{m}{q_i}\right) = m\hat{H}_m$, where $\hat{H}_m = -\sum_{\substack{i \in \{1,\ldots,n\} \\ q_i > 0}} \frac{q_i}{m} \log\left(\frac{q_i}{m}\right)$ is the empirical entropy of the observations [Shannon, 1948]. This is a fundamental lower bound of the total cost.

Theorem 3.3 (Static Optimality Theorem: Version 2)**.** *Let T be a fixed Binary Search Tree and $\ell_i = depth(k_i)$. If we perform m searches among the keys such that $\forall i \in \{1, ..., n\}$, k_i is searched q_i times, then:*

$$\textit{Total Cost of Splay} = \mathcal{O}\left(\sum_{i \in \{1,...,n\}} \ell_i q_i\right)$$

Remark 3.4. The version 2 of the Static Optimality Theorem is weaker (it only applies to fixed Binary Search Trees).

In order to prove the Static Optimality Theorem, we will need a stronger version of the Access Lemma with the following properties:

1. Each node x of T has a weight $w(x)$ such that $w(x) > 0$
2. For each node x of T, $S(x) = \sum_{y \in subtree(x)} w(y)$
3. For each node x of T, $rank(x) = \log_2(S(x))$
4. The potential of the tree is $\Phi(T) = \sum_{x \in T} rank(x)$
5. $unitCost = \#$ rotations

Theorem 3.5 (Access Lemma (stronger version))**.**

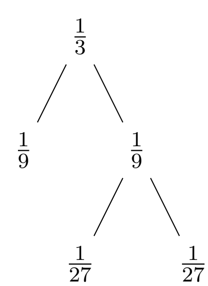
$$AC(splay(x)) \leq 3(rank(root) - rank(x)) + 1$$

The proof is similar to that of the regular Access Lemma

Remark 3.6. 1. This Access Lemma is independent of the weights.

2. $rank(x) - rank(y) = \log_2(\frac{S(x)}{S(y)})$, thus it is independent of the scaling of w as long as $0 < S(x) < \infty$.
3. Must ensure that $\Phi_{begin} - \Phi_{end}$ is small.

Proof. Version 2 of the Static Optimality Theorem (for fixed tree)
Let $\ell(x)$ = depth of x in T, with $\ell(root) = 1$. Let $w(x) = 3^{-l(x)}$.



Note: $S(root) \leq \frac{1}{3} + 2(\frac{1}{3})^2 + 2^2(\frac{1}{3})^3 + \cdots = \frac{1}{3}(1 + \frac{2}{3} + (\frac{2}{3})^2 + \ldots) = \frac{1}{3}(\frac{1}{1-2/3}) = 1$

$$\begin{aligned}
AC(splay(x)) &= \mathcal{O}(1 + \log_2(\frac{S(root)}{S(x)})) \\
&= \mathcal{O}(1 + \log_2(\frac{1}{3^{-l(x)}})) \\
&= \mathcal{O}(1 + \log_2(3^{l(x)})) \\
&= \mathcal{O}(1 + \ell(x))
\end{aligned}$$

$$\begin{aligned}
\text{Total-AC} &= \mathcal{O}(m + \sum_{x \in T} \ell(x) q(x)) \\
&= \mathcal{O}(\sum_{x \in T} \ell(x) q(x))
\end{aligned}$$

Now we can use the amortized cost to find the actual cost: $Cost = AC + \Phi_{begin} - \Phi_{end}$. For any $x \in T$, we have

$$\begin{aligned}
rank_{begin}(x) &= \log_2(S(x)) \leq \log_2(S(root)) \leq \log_2(1) = 0 \\
rank_{end}(x) &= \log_2(S(x)) \geq \log_2(w(x)) = \log_2(3^{-\ell(x)}) = -\ell(x)\log_2(3) \\
\Phi_{begin} - \Phi_{end} &= \sum_{x \in T} rank_{begin}(x) - \sum_{x \in T} rank_{end}(x) \\
&\leq \sum_{x \in T} \ell(x) \log_2(3) \\
&= \mathcal{O}(\sum_{x \in T} \ell(x)) \\
\text{Total Cost} &= \mathcal{O}(\sum_{x \in T} \ell(x) q(x) + \sum_{x \in T} \ell(x)) \\
&= \mathcal{O}(\sum_{x \in T} \ell(x) q(x)) \qquad \text{since } q(x) \geq 1
\end{aligned}$$

□

4 Known Theorems on Splay Trees

4.1 Splays nearby in space

Theorem 4.1 (Scanning Theorem)**.** *Splaying all nodes in any tree T in order (i.e. $splay(k_1), ..., splay(k_n)$) is $\mathcal{O}(n)$ rotations.*

Definition 4.2 (Distance between keys)**.** Let $k_1, ..., k_n$ be our keys. Let $i, j \in \{1, ..., n\}$ such that $j \geq i$.

$$dist(k_j, k_i) = j - i + 1$$

Theorem 4.3 (Static Finger Theorem)**.** *Let f be a fixed key.*

$$AC(splay(x)) = \mathcal{O}(\log(dist(f, x)))$$

Theorem 4.4 (Dynamic Finger Theorem)**.** *Suppose we have request* $x_1, ..., x_m$*.*

$$AC(splay(x_{i+1})) = \mathcal{O}(\log(dist(x_{i+1}, x_i)))$$

4.2 Splays nearby in time

Theorem 4.5 (Working Set Theorem)**.** *Let* $T(x_i) =$ *time since* x_i *was last accessed.*

$$AC(splay(x_i)) = \mathcal{O}(\log T(x_i))$$

References

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423. 3.2

[Sleator and Tarjan, 1985] Sleator, D. D. and Tarjan, R. E. (1985). Self-adjusting binary search trees. *Journal of the ACM (JACM)*, 32(3):652–686. 1.2