CS350 homework 2

Yuchen Hou

September 18, 2015

1 Partition

```
Integer partition(ArrayList<Integer> A, Integer p, Integer q) {
        if p >= q
                return q;
        Integer pivot = A[p]
        Integer index = p;
        while p < q {
                while A[p] < pivot
                        p++;
                while A[q] > pivot
                        q--;
                if p < q
                        swap(A[p], A[q]);
        }
        swap (A[index], A[q]);
        return q;
}
```

2 Insertion sort

This proof is independent of the initial ordering of array A, e.g. whether A is monotonically decreasing or not. In fact, the time complexity of inserting one number into an array with n-a element is always O(n).

2.1 Recursive relation

$T_{insert}(n) = T_{insert}(n-1) + O(n)$

2.2 Statement

$T_{insert}(n) = O(n^2)$

2.3 Proof by induction

Induction hypothesis: $\forall m < n, T_{insert}(m) = O(m^2) < bm^2$

$$\begin{aligned}
T_{insert}(n) &= O((n-1)^2) + O(n) \\
&< b(n-1)^2 + an \\
&= bn^2 + (a-2b)n + b \\
&< bn^2 \text{ // choose a large enough b}
\end{aligned}$$

3 Iqsort

$T_{iq}(n) = T_{partition}(n) + T_{quick}(r-1) + T_{insert}(n-r)$

3.1 Best case: A is already sorted

$$
\begin{aligned}
T_{quick}(r-1) &= O((r-1)^2) \\
T_{insert}(n-r) &= O(n-r) \\
T_{partition}(n) &= O(n) \\
T_{iq}(n) &= \min_r (T_{partition}(n) + T_{quick}(r-1) + T_{insert}(n-r)) \\
&= \min_r (cn + a(r-1)^2 + b(n-r)) \\
&= ((c+b)n - br) \text{ // choose r = 1} \\
&< dn \text{ // choose a large enough d} \\
\implies T_{iq} &= O(n)
\end{aligned}
$$

3.2 Average case

$$
\begin{aligned}
T_{quick}(r-1) &= O((r-1)\log(r-1)) \\
T_{insert}(n-r) &= O((n-r)^2) \\
T_{partition}(n) &= O(n) \\
T_{iq}(n) &= \sum_r (T_{partition}(n) + T_{quick}(r-1) + T_{insert}(n-r))/n \\
&= \sum_r (cn + a(r-1)\log(r-1) + b(n-r)^2)/n \\
&= \int_1^{n+1} (cn + a(r-1)\log(r-1) + b(n-r)^2)/ndr \\
&< an\log_2(n)/2 - an/(4\log_e(2)) - cn \\
&< dn\log(n) \text{ // choose a large enough d} \\
\implies T_{iq} &= O(n\log(n))
\end{aligned}
$$

3.3 Worst case: A is in reverse order

$$
\begin{aligned}
T_{quick}(r-1) &= O((r-1)^2) \\
T_{insert}(n-r) &= O((n-r)^2) \\
T_{partition}(n) &= O(n) \\
T_{iq}(n) &= \max_r (T_{partition}(n) + T_{quick}(r-1) + T_{insert}(n-r)) \\
&= \max_r (cn + a(r-1)^2 + b((n-r)^2) \\
&= \max_r (cn + bn^2 - 2bnr - 2ar + (a+b)r^2) \\
&< dn^2 \text{ // choose a large enough d} \\
\implies T_{iq} &= O(n^2)
\end{aligned}
$$

4 Mixsort

4.1 Best case: A is already sorted

4.1.1 Recursive relation

$$\begin{aligned}
T_{mix}(n) &= T_{partition}(n) + T_{mix}(r-1) + T_{insert}(n-r) \\
T_{partition}(n) &= O(n) \\
T_{insert}(n-r) &= O(n-r)
\end{aligned}$$

4.1.2 Statement

$T_{mix}(n) = O(n)$

4.1.3 Proof by induction

Induction hypothesis: $\forall m < n, T_{mix}(m) = O(m) < am$

$$\begin{aligned}
T_{mix}(n) &= \min_r (T_{partition}(n) + T_{mix}(r-1) + T_{insert}(n-r)) \\
&< \min_r (cn + a(r-1) + b(n-r)) \\
&< cn + b(n-1) \text{ // choose a very large a, r = 1} \\
&< (c+b)n - b \\
&< an \text{ // choose a large enough a} \\
\implies T_{mix}(n) &= O(n)
\end{aligned}$$

4.2 Average case

4.2.1 Recursive relation

$$\begin{aligned}
T_{mix}(n) &= T_{partition}(n) + T_{mix}(r-1) + T_{insert}(n-r) \\
T_{partition}(n) &= O(n) \\
T_{insert}(n-r) &= O((n-r)^2)
\end{aligned}$$

4.2.2 Statement

$T_{mix}(n) = O(n \log n)$

4.2.3 Proof by induction

Induction hypothesis: $\forall m < n, T_{mix}(m) = O(m \log m) < am \log m$

$$\begin{aligned}
T_{mix}(n) &= \sum_r (T_{partition}(n) + T_{mix}(r-1) + T_{insert}(n-r))/n \\
&< \sum_r (cn + a(r-1)\log(r-1) + b(n-r)^2)/n \\
&< \int_1^{n+1} (cn + a(r-1)\log(r-1) + b(n-r)^2)/n \\
&= an\log(n)/2 - an/4\log_e(2) + cn \\
&< an\log(n) \text{ // choose a large enough a} \\
\implies T_{mix} &= O(n\log(n))
\end{aligned}$$

4.3 Worst case: A is in reverse order

4.3.1 Recursive relation

$$\begin{aligned}
T_{mix}(n) &= T_{partition}(n) + T_{mix}(r-1) + T_{insert}(n-r) \\
T_{partition}(n) &= O(n) \\
T_{insert}(n-r) &= O((n-r)^2)
\end{aligned}$$

4.3.2 Statement

$T_{mix}(n) = O(n^2)$

4.3.3 Proof by induction

Induction hypothesis: $\forall m < n, T_{mix}(m) = O(m) < am^2$

$$\begin{aligned}
T_{mix}(n) &= \max_r (T_{partition}(n) + T_{mix}(r-1) + T_{insert}(n-r)) \\
&< \max_r (cn + a(r-1)^2 + b(n-r)^2) \\
&= \max(cn + b(n-1)^2, cn + a(n-1)^2) \\
&= an^2 + (c-2a)n + a \text{ // choose a very large a} \\
&< an^2 \text{ // choose a large enough a} \\
\implies T_{mix} &= O(n^2)
\end{aligned}$$