University of Waterloo
CS 466 — Advanced Algorithm
Spring 2013
Problem Set 9
Siwei Yang - 20258568

1. [14 marks: Mergesort] Consider Mergesort when n is not (necessarily) a power of 2. The method works by (recursively) sorting a subarray of size $\frac{n}{2}$ and one of size $\frac{n}{2}$ and then merging them in n-1 comparisons. A segment of length 1 requires 0 comparisons.

   NOTE: Without loss of generality, assume sorting by asecending order

   - (a) [2 marks] Give a recurrence relation that describes the number of comparisons used, in the worst case, by this method.

   $$C(n) = \begin{cases} C(\lfloor \frac{n}{2} \rfloor) + C(\lceil \frac{n}{2} \rceil) + n - 1, & \text{if } n > 1 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

   - (b) [4 marks] Prove that $n-1$ comparisons are necessary (i.e. you cannot do it in fewer), in the *worst case* for this merge step.

     For any merge step, assume α and β are the two sorted arrays to be merged. Adopt an adversary that select item from the longer array on each comparison for worst case analysis. Then we can infer the following:

     - α and β have respective length $\lfloor \frac{n}{2} \rfloor$ and $\lceil \frac{n}{2} \rceil$ without loss of generality
     - every comparison reduce the length of either α or β by 1
     - before either array is empty, both α and β will have a length of 1
     - when both α and β have a length of 1, the next comparison is the last one needed

     Thus, when both α and β have a length of 1, the number of comparisons already taking place is $n - 1 - 1 = n - 2$. And, the

next comparison finishes the merge which leave the total number of comparison at $n-1$. Therefore, $n-1$ comparisons are necessary in the worst case.

- (c) [4 marks] Prove that Mergesort, as described above, takes $n * \lceil \lg n \rceil - 2^{\lceil \lg n \rceil} + 1$ comparisons in the worst case.

  First consider n that is a power of 2.

  - The expression converts to $n * \lg n - n + 1$.
  - base case $n * \lg n - n + 1$ holds for $n = 2^1$ by definition of C.
  - induction Given $n * \lg n - n + 1$ holds for $n = 2^i$, by definition of C, $C(2^{i+1}) = C(2^i) + C(2^i) + 2^{i+1} - 1 = 2 * 2^i * i - 2 * 2^i + 2 + 2^{i+1} - 1 = 2^{i+1} * (i+1) - 2^{i+1} + 1$. Thus, $n * \lg n - n + 1$ holds for $n = 2^{i+1}$.
  - conclusion Therefore, $n * \lg n - n + 1$ holds for all power of 2. Equivalently, $n * \lceil \lg n \rceil - 2^{\lceil \lg n \rceil} + 1$ holds for all power of 2.

  Then, for number k that is not a power of 2, consider the smallest power of 2 that's greater than k. Assume $k = 2^i - j > 2^{i-1}$:

  - $C(2^i) = 2^i * i - 2^i + 1$.
  - consider each item taken from the array, i comparisons are saved as the recursion depth is i(until the total number of item fall to 2^{i-1}).
  - given $k = 2^i - j > 2^{i-1}$, $C(k) = C(2^i) - j * i = (2^i - j) * i - 2^i + 1$

  Since $\lceil \lg k \rceil = i$, we have $C(k) = k * \lceil \lg k \rceil - 2^{\lceil \lg k \rceil} + 1$. **Therefore, we have** $n * \lceil \lg n \rceil - 2^{\lceil \lg n \rceil} + 1$ **for all** $n \geq 2$.

- (d) [4 marks] The *expected* number of comparisons for this method (over all possible permutations of the input) is a little $\theta(n)$ better. Prove it. (You do not have to deal with the exact constant in this $\theta(n)$ term.)

  Consider that the number of comparisons at each merge is determined by the size of maximum suffix of either arrays where the items in the suffix is greater than any item not in the suffix. Therefore, we can use combinatorics to analysis the probability of number of comparisons on each merge.

  Assume α and β are the two sorted arrays to be merged with respective length m and n, and each relative ordering is equally

likely. Also note that the relative orderings are all independent on each merge steps.

- The total number of unique relative ordering between all items are $\binom{m+n}{n} = \binom{m+n}{m}$.
- The total number of unique relative ordering where the maximum suffix size is **k or more** amounts to $\binom{m+n-k}{n} + \binom{m+n-k}{m}$ where the k greatest items are group togather and the rest $m+n-k$ are split into $(m, n-k)$ or $(n, m-k)$.
- Each relative ordering where the maximum suffix size is **k or more**, saves $k-1$ comparisons from the worst case.
- The total number of comparisons saved from the worst case is bounded by the summation $S = \sum_{k>1}^{m} (k-1) * \binom{m+n-k}{n} + \sum_{k>1}^{n} (k-1) * \binom{m+n-k}{m}$.

We conclude the average number of comparisons saved on each merge step is upper bounded by:

$$\sum_{k>1}^{m} \frac{(k-1) * \binom{m+n-k}{n}}{\binom{m+n}{n}} + \sum_{k>1}^{n} \frac{(k-1) * \binom{m+n-k}{m}}{\binom{m+n}{m}} \tag{2}$$

Knowing that this expression is bounded by a constant, c, and there are at most n merges in total, the total numbers of comparions saved is upper bounded by $c * n$.

Let's look at the lower bound now:

- Consider the special case 1 where α and β are the two sorted arrays to be merged with respective length 2 and 1. The total number of unique relative ordering where the maximum suffix size is **2 or more** amounts to $\binom{1}{1} + \binom{1}{2} = 1$. Thus average number of comparisons saved from worst case is lower bounded by $\frac{1}{\binom{1+2}{1}} = \frac{1}{3}$.
- Consider the special case 2 where α and β are the two sorted arrays to be merged with respective length 2 and 2. The total number of unique relative ordering where the maximum suffix size is **2 or more** amounts to $\binom{2}{2} + \binom{2}{2} = 2$. Thus average number of comparisons saved from worst case is lower bounded by $\frac{2}{\binom{2+2}{2}} = \frac{1}{3}$.

We know that each of the n items have either gone through a merge of $(2, 1)$ or $(2, 2)$. Therefore, assuming there are i merges of $(2, 1)$ and j merges of $(2, 2)$, we have the following:

$$i * 3 + j * 4 \geq n \tag{3}$$

which gives us $i + j \geq \frac{n}{4}$. **Since either of the merges saves $\frac{1}{3}$ comparisons, the total numbers of comparions saved is lower bounded by $\frac{n}{4} * \frac{1}{3} = \frac{n}{12}$.**
Therefore, **the comparions saved from the worst case is bound by θn.**