CS 466/666
Matula's 2^{nd} Largest Algorithm

We have mentioned that even the worst case lower bound of 1¾ n - O(1) comparisons for finding the median is higher than the expected case algorithm that uses 1½ n - o(n). A similar situation exists for the problem of finding the second largest of n numbers, though the difference is in the second order term. We noted that the technique of repeated pairing to find the maximum, followed by finding the largest of the lg n -1 elements that lost directly to the maximum, leads to a worst case optimal n + lg n + O(1) method.
Sometime in the 1970's David Matula suggested a "better" method for the expected case. (It appeared as a tech report which he is sending me, but does not appear to have been published otherwise). The improvement is to reduce the lg n to a Θ(lg lg n) term, so the improvement is mostly of interest in showing how worst case can be higher than expected case.

First take a random sample of size s = n/lg n, and find its maximum by repeated pairings.
Set cmax = sample maximum; and ctwo = the element that lost the last comparison to it.
For each of the rest of elements, compare the element with ctwo if it is larger with cmax. Adjust cmax and ctwo so they are the two largest values seen in this scan and the start values.

The algorithm works properly if a value larger than the sample maximum is found among the remainder of the elements. So with probability 1 – s/n = 1 – 1/lg n, the method uses a number of comparisons equal to n-1 plus the number of times a new value in the top two occurs after the first s. The probability that the i^{th} value is in the top 2 of the first i values is 2/i, so this gives an expected 2/i comparisons of element i with cmax. This runs from position s+1 to position n, so approximating this sum with an integral we have a total of 2 ln (n/s)+O(1) = 2 lnln n +O(1) "second comparisons".

The "other case" is that the sample max is the set maximum (probability s/n). In this situation, the second largest is the maximum of the final ctwo and those that lost to the sample max in the pairing (excluding the last comparison within the sample). That means another lg n – 2 comparisons.

Our expected total, then, is n + 2 lnln n + (lg n -2)(n/(n lg n) = n + 2 lnln n +O(1).